

A COALGEBRAIC APPROACH TO QUANTITATIVE LINEAR TIME LOGICS

CORINA CÎRSTEA

University of Southampton, UK
e-mail address: cc2@ecs.soton.ac.uk

ABSTRACT. We define quantitative fixpoint logics for reasoning about linear time properties of states in systems with branching behaviour. We model such systems as coalgebras whose type arises as the composition of a branching monad with one or more polynomial endofunctors on the category of sets. The domain of truth values for our logics is determined by the choice of branching monad, as is the special modality used to abstract away branching in the semantics of the logics. To justify our (canonical) choice of syntax and semantics for the logics, we prove the equivalence between their step-wise semantics and an alternative path-based semantics for the fixpoint-free fragments of the logics. Instances of our logics support reasoning about the possibility, probability or minimal cost of exhibiting a given linear time property. We conclude with two examples of logics that have a linear time flavour, namely a logic for reasoning about resource usage in infinitely-running computations, and a logic for reasoning about component interaction. Some non-canonical choices are made for the modalities employed by these logics, and in the latter case this prevents the existence of a path-based semantics. Yet, both logics have clear practical relevance, and illustrate the flexibility of our approach.

1. INTRODUCTION

Linear time temporal logics such as LTL or the linear time μ -calculus (see e.g. [Dam92]), originally interpreted over non-deterministic systems, have been adapted and used successfully in model checking both non-deterministic and probabilistic systems (see e.g. [BK08]). These logics share the same notion of linear time behaviour, but depending on the type of branching present in the underlying models (non-deterministic or probabilistic), have either a qualitative or a quantitative interpretation (with $\{0, 1\}$ and respectively the unit interval $[0, 1]$ as domains of truth values). Despite commonalities between these logics and their associated (automata-based) verification techniques, a general and uniform account of linear time logics and their use in formal verification is still missing.

The present paper attempts to fill this gap, by building on recent work on a coalgebraic account of linear time behaviour in systems with branching [Cîr17]. We model systems as coalgebras whose type incorporates both branching and linear behaviour, and define quantitative fixpoint logics that are *parametric* in both the branching type and the transition type. The branching type determines the domain of truth values for the logics, and dictates

2012 ACM CCS: [Theory of computation]: Logic and verification; Modal and temporal logics;

Key words and phrases: coalgebra; fixpoint logic; quantitative logic; linear time logic.

the choice of modality used in the semantics of the logics to abstract away branching, as well as the choice of propositional operators employed by the logics. The transition type canonically induces a notion of observable linear behaviour, together with associated linear time modalities. The canonical choices made in the semantics of the logics allow for an alternative path-based semantics to be defined for the fixpoint-free fragment of our logics, and proved equivalent to the original, step-wise semantics. Moreover, this equivalence result generalises to certain non-canonical choices of modalities for the logics. In addition to non-deterministic and probabilistic branching, our logics also instantiate to systems with weighted branching.

A more detailed outline of our approach and results is given below:

- We model systems as coalgebras of endofunctors obtained by composing a *branching monad* $\mathsf{T} : \mathsf{Set} \rightarrow \mathsf{Set}$ with a *polynomial endofunctor* $F : \mathsf{Set} \rightarrow \mathsf{Set}$. The branching monad arises from a partial commutative semiring $S = (S, +, 0, *, 1)$, whose carrier provides the domain of truth values for our logics. The semantics of the logics therefore requires *quantitative predicate liftings* to interpret both the linear time modalities and the (hidden) branching modality. While canonical choices exist for both types of modalities, non-canonical choices are also allowed.
- A key feature of our logics is their *step-wise* semantics; that is, the interpretation of a formula is defined by repeatedly unfolding the coalgebra structure, as required by the structure of the formula. This is different from logics such as LTL or the linear time μ -calculus, where the interpretation of a formula on a state depends on its interpretation on the (infinite) paths from that state. In the case of probabilistic branching, defining such a path-based semantics requires the use of measure theory, whereas our step-wise semantics avoids this, while achieving a similar interpretation. In the case of weighted branching, we are not aware of weighted linear time logics similar to the ones proposed here, but speculate (based on related work on weighted Büchi automata [DR06], see also Section 5) that different technical machinery would be required to define a path-based semantics for such logics. In particular, this makes the quest for a *uniform* path-based semantics for the full fixpoint logics a difficult one.
- To substantiate the canonical choices made in the definition of our logics, we define an alternative path-based semantics for the *fixpoint-free fragments* of the logics, which we then prove equivalent to the step-wise semantics. Conditions under which similar, non-canonical logics enjoy the same equivalence result are also studied. Both the definition of the path-based semantics and its equivalence with the step-wise semantics rely on a reformulation of our logics using the framework of dual adjunctions (see e.g. [BK05] for the use of dual adjunctions in defining logics for coalgebras). This categorical formulation allows us to isolate the key requirement for the equivalence result (Lemma 5.12), namely a certain commutativity property between linear and branching time modalities. We also note that the existence of a strong connection between our logics and quantitative parity automata, recently studied in [CSH] (see also Example 4.6), mitigates the lack of a path-based semantics for the *full* fixpoint logics, by providing additional evidence that our logics subsume known linear time logics. This is discussed further in Section 5.
- While for the majority of the paper we consider coalgebras of type $\mathsf{T} \circ F$ as mentioned above, we conclude the paper by moving to more general coalgebraic types, namely $F_1 \circ \mathsf{T} \circ \dots \circ F_{n-1} \circ \mathsf{T} \circ F_n$, with T as before and F_1, \dots, F_n arbitrary endofunctors. We use two examples to illustrate how a modular approach to defining logics for such complex coalgebraic types can yield logics with a linear time flavour. Only the first of these logics

admits an equivalent path-based semantics, as the modalities employed by the second one do not commute as required by our equivalence result.

The remainder of the paper is structured as follows. Section 2 introduces partial semiring monads and summarises our previous coalgebraic account of linear time behaviour in systems with branching [C  r17]. Section 3 defines semiring-valued predicate liftings, which are then used in Section 4 to define quantitative, linear time fixpoint logics for coalgebras with branching. Section 5 provides an alternative path-based semantics for the fixpoint-free fragment of the logics, and proves its equivalence to the step-wise semantics. Section 6 describes how the logics presented earlier can be extended to coalgebraic types arising as arbitrary compositions of a branching monad with several endofunctors, and describes two specific, non-canonical instances of such logics, one for reasoning about resource-aware, infinitely running computations, and another for reasoning quantitatively about component interaction. Section 7 concludes with a discussion of ongoing and future work.

This paper is based on the conference papers [C  r14, C  r15], which it extends by generalising the type of coalgebraic functors used to model systems with branching and the predicate liftings used in the semantics of the logics. These generalisations substantially widen the applicability of the logics, as emphasised in Section 6. The logics presented here are a slight variant of those in [C  r15], see Remark 2.4. Furthermore, the proof of equivalence between the step-wise semantics and the path-based semantics now applies to the full fixpoint-free fragment of the logics, and not just to the so-called *uniform* fragment, as considered in [C  r15].

2. PRELIMINARIES

2.1. Monads and Partial Semirings. In what follows, we use monads (T, η, μ) on **Set** (where $\eta : \text{Id} \Rightarrow T$ and $\mu : T \circ T \Rightarrow T$ are the *unit* and *multiplication* of T) to capture branching in coalgebraic types. The specific nature of the monads we consider makes them *strong* and *commutative*. A *strong monad* comes equipped with a *strength map* $\text{st}_{X,Y} : X \times TY \rightarrow T(X \times Y)$, natural in X and Y and subject to coherence conditions w.r.t. η and μ (see e.g. [Jac16, Chapter 5] for details). For such a monad, one can also define a *swapped strength map* $\text{st}'_{X,Y} : TX \times Y \rightarrow T(X \times Y)$ by:

$$TX \times Y \xrightarrow{\text{tw}_{TX,Y}} Y \times TX \xrightarrow{\text{st}_{Y,X}} T(Y \times X) \xrightarrow{T\text{tw}_{Y,X}} T(X \times Y)$$

where $\text{tw}_{X,Y} : X \times Y \rightarrow Y \times X$ is the *twist map* taking $(x, y) \in X \times Y$ to (y, x) . *Commutative monads* are strong monads where the maps $\mu_{X,Y} \circ T\text{st}'_{X,Y} \circ \text{st}_{TX,Y} : TX \times TY \rightarrow T(X \times Y)$ and $\mu_{X,Y} \circ T\text{st}_{X,Y} \circ \text{st}'_{TX,Y} : TX \times TY \rightarrow T(X \times Y)$ coincide, yielding a *double strength map* $\text{dst}_{X,Y} : TX \times TY \rightarrow T(X \times Y)$ for each choice of sets X, Y .

The monads considered in this paper arise from partial commutative semirings.

Definition 2.1. A *partial commutative semiring* is a tuple $S := (S, +, 0, \bullet, 1)$ with $(S, +, 0)$ a partial commutative monoid and $(S, \bullet, 1)$ a commutative monoid, with \bullet distributing over $+$; that is, for all $s, t, u \in S$, $s \bullet 0 = 0$, and whenever $t + u$ is defined, then so is $s \bullet t + s \bullet u$ and moreover $s \bullet (t + u) = s \bullet t + s \bullet u$.

The addition operation of any partial commutative semiring induces a pre-order relation $\sqsubseteq \subseteq S \times S$, given by

$$x \sqsubseteq y \text{ if and only if there exists } z \in S \text{ such that } x + z = y \quad (2.1)$$

for $x, y \in S$. It follows immediately from the axioms of a partial commutative semiring (see [Cîr17]) that \sqsubseteq has $0 \in S$ as bottom element and is preserved by \bullet in each argument.

Assumption 2.2. *We assume that \sqsubseteq is an ω -chain complete as well as ω^{op} -chain complete partial order, and has the unit 1 of \bullet as top element.*

Example 2.3. In what follows, we consider the *boolean semiring* $(\{0, 1\}, \vee, 0, \wedge, 1)$, the (partial) *probabilistic semiring* $([0, 1], +, 0, *, 1)$, the *tropical semiring* $\mathbb{N} = (\mathbb{N}^\infty, \min, \infty, +, 0)$ and its bounded variants $S_B = ([0, B]^\infty, \min, \infty, +_B, 0)$ with $B \in \mathbb{N}$, where for $m, n \in [0, B]^\infty$, $m +_B n = \begin{cases} m + n, & \text{if } m + n \leq B \\ \infty, & \text{otherwise} \end{cases}$. The associated pre-orders are \leq on $\{0, 1\}$ and $[0, 1]$, and \geq on \mathbb{N}^∞ and $[0, B]^\infty$. All these pre-orders satisfy Assumption 2.2.

A partial commutative semiring S induces a *semiring monad* $(\mathsf{T}_S, \eta, \mu)$ with

$$\mathsf{T}_S(X) = \{ \varphi : X \rightarrow S \mid \text{supp}(\varphi) \text{ is finite, } \sum_{x \in \text{supp}(\varphi)} \varphi(x) \text{ is defined} \}$$

$$\eta_X(x)(y) = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise} \end{cases}, \quad \mu_X(\Phi)(x) = \sum_{\varphi \in \text{supp}(\Phi)} \Phi(\varphi) \bullet \varphi(x)$$

where $\text{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$ is the *support* of φ . We use the formal sum notation $\sum_{i \in I} c_i x_i$, with I finite, to denote the element of $\mathsf{T}_S(X)$ mapping x_i to c_i for $i \in I$, and $x \notin \{x_i \mid i \in I\}$ to 0.

The monad T_S above is *strong* and *commutative*, with *strength* $\text{st}_{X,Y} : X \times \mathsf{T}_S Y \rightarrow \mathsf{T}_S(X \times Y)$ and *double strength* $\text{dst}_{X,Y} : \mathsf{T}_S X \times \mathsf{T}_S Y \rightarrow \mathsf{T}_S(X \times Y)$ given by

$$\text{st}_{X,Y}(x, \psi)(z, y) = \begin{cases} \psi(y) & \text{if } z = x \\ 0 & \text{otherwise} \end{cases}, \quad \text{dst}_{X,Y}(\varphi, \psi)(z, y) = \varphi(z) \bullet \psi(y)$$

We also note that $\mathsf{T}_S 1 = S$. In what follows, we will use $\mathsf{T}_S 1$ and S interchangeably, in particular $\mathsf{T}_S 1$ will be used in contexts where its (free) T_S -algebra structure (given by $\mu_1 : \mathsf{T}_S^2 1 \rightarrow \mathsf{T}_S 1$) is relevant.

The relationship between monads and partial semirings was thoroughly studied in [CJ13, Cîr17]. We use semiring monads to model branching, with the semirings in Example 2.3 accounting for finite non-deterministic, probabilistic and weighted branching. In the latter case, we think of the weights as costs associated to single computation steps, with the bounded variant of the tropical semiring capturing the existence an upper limit on the maximum allowable costs.

Remark 2.4. Our earlier work [Cîr14, Cîr15] was parameterised by a so-called *partially additive monad*. The connection with the partial semiring monads used here is as follows: any commutative, partially additive monad which is, in addition, finitary, is isomorphic to a partial semiring monad (see [Cîr15, Remark 2.4]). While our earlier work also covers the unbounded powerset monad (as an additive monad), we are not aware of any other non-finitary monads to which the results in [Cîr14, Cîr15] apply. For this reason, the present paper restricts attention to semiring monads.

2.2. Generalised Relations and Relation Lifting. Throughout this section we fix a partial commutative semiring $(S, +, 0, \bullet, 1)$ satisfying Assumption 2.2.

We let Rel denote the category¹ with objects given by triples (X, Y, R) , where $R : X \times Y \rightarrow S$ is an S -valued relation, and with arrows from (X, Y, R) to (X', Y', R') given by pairs of functions (f, g) as below, such that $R \sqsubseteq R' \circ (f \times g)$:

$$\begin{array}{ccc} X \times Y & \xrightarrow{f \times g} & X' \times Y' \\ R \downarrow & \sqsubseteq & \downarrow R' \\ S & \xlongequal{\quad} & S \end{array}$$

Here, the order \sqsubseteq on S has been extended pointwise to S -valued relations with the same carrier. We write $q : \text{Rel} \rightarrow \text{Set} \times \text{Set}$ for the functor taking (X, Y, R) to (X, Y) and (f, g) to itself. It follows easily that q is a fibration, with reindexing functors $(f, g)^* : \text{Rel}_{X', Y'} \rightarrow \text{Rel}_{X, Y}$ taking $R' : X' \times Y' \rightarrow S$ to $R' \circ (f \times g) : X \times Y \rightarrow S$. We also write $\text{Rel}_{X, Y}$ for the *fibre over* (X, Y) , i.e. the subcategory of Rel with objects given by S -valued relations over $X \times Y$ and arrows given by $(1_X, 1_Y)$.

[C  r17] shows how to canonically lift *polynomial* endofunctors on Set (that is, endofunctors constructed from identity and constant functors using *finite* products and set-indexed coproducts) to the category of S -valued relations. The *relation lifting of* $F : \text{Set} \rightarrow \text{Set}$ is an endofunctor $\text{Rel}(F) : \text{Rel} \rightarrow \text{Rel}$ making the following diagram commute:

$$\begin{array}{ccc} \text{Rel} & \xrightarrow{\text{Rel}(F)} & \text{Rel} \\ q \downarrow & & \downarrow q \\ \text{Set} \times \text{Set} & \xrightarrow{F \times F} & \text{Set} \times \text{Set} \end{array}$$

The definition of $\text{Rel}(F)$ is by structural induction on F and makes use of the partial semiring structure on S :

- If $F = \text{Id}$, $\text{Rel}(F)$ takes an S -valued relation to itself.
- If $F = C$, $\text{Rel}(F)$ takes an S -valued relation to the *equality relation* $\text{Eq}(C) : C \times C \rightarrow S$ given by

$$\text{Eq}_C(c, c') = \begin{cases} \top & \text{if } c = c' \\ \perp & \text{otherwise} \end{cases}.$$

- If $F = F_1 \times F_2$, $\text{Rel}(F)$ takes an S -valued relation $R : X \times Y \rightarrow S$ to:

$$(F_1 X \times F_2 X) \times (F_1 Y \times F_2 Y) \xrightarrow{\langle \pi_1 \times \pi_1, \pi_2 \times \pi_2 \rangle} (F_1 X \times F_1 Y) \times (F_2 X \times F_2 Y) \xrightarrow{\text{Rel}(F_1)(R) \times \text{Rel}(F_2)(R)} S \times S \xrightarrow{\bullet} S.$$

The functoriality of this definition follows from the preservation of \sqsubseteq by •.

- if $F = \coprod_{i \in I} F_i$, $\text{Rel}(F)(R) : (\coprod_{i \in I} F_i X) \times (\coprod_{i \in I} F_i Y) \rightarrow S$ is defined by case analysis:

$$\text{Rel}(F)(R)(\iota_i(u), \iota_j(v)) = \begin{cases} \text{Rel}(F_i)(R)(u, v) & \text{if } i = j \\ \perp & \text{otherwise} \end{cases}$$

for $i, j \in I$, $u \in F_i X$ and $v \in F_j Y$.

¹To keep notation simple, the dependency on S is left implicit.

It follows immediately from the above definition that $q \circ \text{Rel}(F) = (F \times F) \circ q$.

A special relation lifting, called *extension lifting* is defined in [Cîr17] for any commutative, partially additive monad \mathbb{T} . The extension lifting $E_{\mathbb{T}} : \text{Rel} \rightarrow \text{Rel}$ lifts the endofunctor $\mathbb{T} \times \text{Id}$ to Rel

$$\begin{array}{ccc} \text{Rel} & \xrightarrow{E_{\mathbb{T}}} & \text{Rel} \\ q \downarrow & & \downarrow q \\ \text{Set} \times \text{Set} & \xrightarrow{\mathbb{T} \times \text{Id}} & \text{Set} \times \text{Set} \end{array}$$

in a canonical way. In the special case when \mathbb{T} is the partial semiring monad \mathbb{T}_S , the extension lifting takes $R : X \times Y \rightarrow S$ to the relation $E_{\mathbb{T}}(R) : \mathbb{T}X \times Y \rightarrow S$ given by

$$E_{\mathbb{T}}(R)\left(\sum_{i \in I} c_i x_i, y\right) = \sum_{i \in I} c_i \bullet R(x_i, y)$$

with $c_i \in S$ and $x_i \in X$ for $i \in I$, and $y \in Y$. We note here that the definedness of $\sum_{i \in I} c_i$ together with the unit 1 of S being the top element of (S, \sqsubseteq) imply that $\sum_i c_i \bullet R(x_i, y)$ is itself defined.

Remark 2.5. The more general definition of relation lifting maps $R : X \times Y \rightarrow \mathbb{T}1$ to

$$\mathbb{T}X \times Y \xrightarrow{\text{st}'_{X,Y}} \mathbb{T}(X \times Y) \xrightarrow{\mathbb{T}(R)} \mathbb{T}^2 1 \xrightarrow{\mu_1} \mathbb{T}1$$

It is straightforward to check that, for partial semiring monads, the two definitions coincide.

2.3. Linear Time Behaviour via Relation Lifting. We now summarise the definition of the linear time behaviour of a state in a coalgebra with branching, as proposed in [Cîr17]. The approach in loc. cit. applies to coalgebras of functors obtained as arbitrary compositions of a single partially additive monad and a finite number of polynomial endofunctors on Set . For simplicity, here we restrict attention to compositions of type $\mathbb{T}_S \circ F$, with S a partial commutative semiring. Thus, we model systems with branching as $\mathbb{T}_S \circ F$ -coalgebras on Set , where the partial semiring monad $\mathbb{T}_S : \text{Set} \rightarrow \text{Set}$ specifies the type of branching, and the polynomial endofunctor $F : \text{Set} \rightarrow \text{Set}$ specifies the structure of individual transitions.

Given an arbitrary endofunctor $F : \text{Set} \rightarrow \text{Set}$, an F -coalgebra is given by a pair (C, γ) with C a set (of states), and $\gamma : C \rightarrow FC$ a function describing the one-step evolution of the states. The notion of *coalgebraic bisimulation* provides a canonical and uniform observational equivalence relation between states of F -coalgebras. One of the many (and largely equivalent) definitions of bisimulation involves lifting the endofunctor F to the category of two-valued relations (obtained in our setting by taking $S = (\{0, 1\}, +, 0, *, 1)$). A similar approach is taken in [Cîr17] to define the extent to which a state in a coalgebra with branching can exhibit a given linear time behaviour. The definition in loc. cit. differs from the relational definition of bisimulation (for which we refer the reader to [Jac16]) in two ways: (i) S -valued relations are used in place of standard relations, and (ii) the relation lifting employed also involves the extension lifting $E_{\mathbb{T}}$ defined earlier, as the goal is to relate branching time and linear time behaviours, as opposed to behaviours with the same coalgebraic type.

We write (Z, ζ) for the final F -coalgebra. This provides a natural choice as domain of observable linear time behaviours. We will also refer to the elements of Z as *maximal traces*.

The definition of the linear time behaviour of states in coalgebras with branching (below) is inspired by a characterisation of coalgebraic bisimilarity (i.e. the largest bisimulation) between states of coalgebras of the same type as the greatest fixpoint of a monotone operator on the category of standard relations.

Definition 2.6 ([C  r17]). The *linear time behaviour* of a state in a $\mathsf{T}_S \circ F$ -coalgebra (C, γ) is the greatest fixpoint of the operator O on $\mathsf{Rel}_{C,Z}$ given by the composition

$$\mathsf{Rel}_{C,Z} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FC,FZ} \xrightarrow{\mathsf{E}_\mathsf{T}} \mathsf{Rel}_{\mathsf{T}(FC),FZ} \xrightarrow{(\gamma \times \zeta)^*} \mathsf{Rel}_{C,Z} \quad (2.2)$$

Monotonicity of the operator O above is an immediate consequence of the functoriality of $\mathsf{Rel}(F)$, E_T and $(\gamma \times \delta)^*$. The existence of a greatest fixpoint for O is then guaranteed by the following standard result on the existence of fixpoints in chain-complete partial orders, applied to the *dual* of the order \sqsubseteq .

Theorem 2.7 ([DP02, Theorem 8.22]). *Let P be a complete partial order and let $\mathsf{O} : P \rightarrow P$ be order-preserving. Then O has a least fixpoint.*

- Examples 2.8.** (1) For $S = (\{0, 1\}, \vee, 0, \wedge, 1)$, the greatest fixpoint of O relates a state c in a $\mathsf{T}_S \circ F$ -coalgebra (C, γ) with a maximal trace $z \in Z$ iff there exists a sequence of choices in the unfolding of γ starting from c that results in an F -behaviour bisimilar to z .
- (2) For $S = ([0, 1], +, 0, *, 1)$, the greatest fixpoint of O yields, for each state in a $\mathcal{S} \circ F$ -coalgebra and each maximal trace, the accumulated probability of this trace being exhibited (across all branches). Here we note that, for *infinite* maximal traces, the associated probability is often 0. One may argue that this has limited usefulness, and that a measure-theoretic definition would in this case be more appropriate. The logics defined later in this paper do not suffer from a similar issue, as they allow reasoning about the probability of exhibiting linear time behaviours that belong to a certain specified *set* of traces.
- (3) For $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, the greatest fixpoint of O maps a pair (c, z) , with c a state in a $\mathsf{T}_S \circ F$ -coalgebra and z a maximal trace, to the minimal cost of exhibiting z from c . Intuitively, this is computed by adding the weights of individual transitions along the same branch, and minimising this sum across the various branches. Again, such minimal costs are often infinite for *infinite* maximal traces, but in this case this simply reflects the fact that infinitely running systems have infinite costs. Section 6.1 will show how resource replenishment can be modelled in our setting through a change of the underlying coalgebraic type.

3. SEMIRING-VALUED PREDICATES AND PREDICATE LIFTING

The standard approach to defining the semantics of modal and fixpoint logics involves interpreting formulas as predicates over the state space of the system of interest. In the coalgebraic approach to modal logic, individual modal operators are interpreted using so called *predicate liftings* [Pat03]. In order to follow the same approach for *quantitative, linear time* logics, we will work with predicates valued in the partial commutative semiring $(S, +, 0, \bullet, 1)$ used to model branching.

We let \mathbf{Pred} denote the category with objects given by pairs (X, P) with $P : X \rightarrow S$ an S -valued predicate, and arrows from (X, P) to (X', P') given by functions $f : X \rightarrow X'$ such that $P \sqsubseteq P' \circ f$:

$$\begin{array}{ccc} X & \xrightarrow{f} & X' \\ P \downarrow & \sqsubseteq & \downarrow P' \\ S & \xlongequal{\quad} & S \end{array}$$

As with S -valued relations, we obtain a fibration $p : \mathbf{Pred} \rightarrow \mathbf{Set}$, with p taking (X, P) to X . The fibre over X is denoted \mathbf{Pred}_X , and the reindexing functor $f^* : \mathbf{Pred}_{X'} \rightarrow \mathbf{Pred}_X$ takes $P' : X' \rightarrow S$ to $P' \circ f : X \rightarrow S$.

The next definition generalises predicate liftings as used in the semantics of coalgebraic modal logics [Pat03].

Definition 3.1. An $(S$ -valued) *predicate lifting* of arity $n \in \omega$ for an endofunctor $F : \mathbf{Set} \rightarrow \mathbf{Set}$ is a functor $L : \mathbf{Pred}^n \rightarrow \mathbf{Pred}$ making the following diagram commute:

$$\begin{array}{ccc} \mathbf{Pred}^n & \xrightarrow{L} & \mathbf{Pred} \\ p \downarrow & & \downarrow p \\ \mathbf{Set} & \xrightarrow{F} & \mathbf{Set} \end{array}$$

where the category \mathbf{Pred}^n has objects given by tuples (X, P_1, \dots, P_n) with $P_i : X \rightarrow S$ for $i \in \{1, \dots, n\}$, and arrows from (X, P_1, \dots, P_n) to (X', P'_1, \dots, P'_n) given by functions $f : X \rightarrow X'$ such that $P_i \sqsubseteq P'_i \circ f$ for all $i \in \{1, \dots, n\}$.

We now restrict attention to polynomial functors $F : \mathbf{Set} \rightarrow \mathbf{Set}$, and show how to define a canonical *set* of predicate liftings for F by induction on its structure. The next definition exploits the observation that any polynomial endofunctor is naturally isomorphic to a coproduct of finite (including empty) products of identity functors.

Definition 3.2 (Canonical predicate liftings). Let $F = \coprod_{i \in I} \mathbf{Id}^{j_i}$, with $j_i \in \omega$ for $i \in I$. The set of predicate liftings $\Lambda = \{L_i \mid i \in I\}$ has elements $L_i : \mathbf{Pred}^{j_i} \rightarrow \mathbf{Pred}$ with $i \in I$ given by:

$$(L_i)_X(P_1, \dots, P_{j_i})(f) = \begin{cases} P_1(x_1) \bullet \dots \bullet P_{j_i}(x_{j_i}), & \text{if } f = (x_1, \dots, x_{j_i}) \in \iota_i(\mathbf{Id}^{j_i}) \\ 0 & \text{otherwise} \end{cases}.$$

The functoriality of this definition follows from the preservation of \sqsubseteq by \bullet .

Example 3.3. For $F = 1 + A \times \mathbf{Id} \times \mathbf{Id} \simeq 1 + \coprod_{a \in A} \mathbf{Id} \times \mathbf{Id}$, F -coalgebras are binary trees with internal nodes labelled by elements of A . Definition 3.2 yields a nullary predicate lifting L_0 and an A -indexed set of binary predicate liftings $(L_a)_{a \in A}$:

$$\begin{aligned} L_0(f) &= \begin{cases} 1, & \text{if } f = \iota_1(*) \\ 0, & \text{otherwise} \end{cases} \\ (L_a)_X(P_1, P_2)(f) &= \begin{cases} P_1(x_1) \bullet P_2(x_2), & \text{if } f = \iota_a(x_1, x_2) \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

Remark 3.4. One can also define a *single*, unary predicate lifting $\mathbf{Pred}(F)$ for each polynomial functor $F : \mathbf{Set} \rightarrow \mathbf{Set}$, again by induction on the structure of F :

- If $F = \text{Id}$, $\text{Pred}(F)$ takes an S -predicate to itself.
- If $F = 1$, $\text{Pred}(F)$ takes an S -predicate to the predicate $* \mapsto 1$.
- If $F = F_1 \times F_2$, $\text{Pred}(F)(P) : F_1 X \times F_2 X \rightarrow S$ is given by

$$\text{Pred}(F)(P)(f_1, f_2) = \text{Pred}(F_1)(P)(f_1) \bullet \text{Pred}(F_2)(P)(f_2), \quad \text{for } P : X \rightarrow S.$$

- if $F = \coprod_{i \in I} F_i$, $\text{Pred}(F)(P) : \coprod_{i \in I} F_i X \rightarrow S$ is given by

$$\text{Pred}(F)(P)(\iota_i(f_i)) = \text{Pred}(F_i)(P)(f_i) \quad \text{for } P : X \rightarrow S, i \in I \text{ and } f_i \in F_i X.$$

The above definition, which follows [HJ98], yields logics with limited expressive power. We will show later how $\text{Pred}(F)$ can provide a coinductive interpretation of truth in a system with branching.

Example 3.5. Let $F : \text{Set} \rightarrow \text{Set}$ be as in Example 3.3. Then $\text{Pred}(F)$ is given by

$$\text{Pred}(F)(P)(\iota_1(*)) = 1 \quad \text{Pred}(F)(P)(\iota_a(x_1, x_2)) = P(x_1) \bullet P(x_2)$$

As can be seen, the resulting unary modality requires the same property (P) to hold on both the left- and the right subtree.

As we are interested in *linear time* logics, a special *extension lifting*, akin to the extension lifting of Section 2.2, will be used to abstract away branching.

Definition 3.6. Let $(S, +, 0, \bullet, 1)$ be a partial commutative semiring with associated monad T_S . The *extension lifting* $E_{T_S} : \text{Pred} \rightarrow \text{Pred}$ is the lifting of $T_S : \text{Set} \rightarrow \text{Set}$ to Pred

$$\begin{array}{ccc} \text{Pred} & \xrightarrow{E_{T_S}} & \text{Pred} \\ p \downarrow & & \downarrow p \\ \text{Set} & \xrightarrow{T_S} & \text{Set} \end{array}$$

which takes $P : X \rightarrow S$ to the predicate $E_{T_S}(P) : T_S X \rightarrow S$ given by

$$\sum_{i \in I} c_i x_i \mapsto \sum_{i \in I} c_i \bullet P(x_i)$$

with $c_i \in S$ for $i \in I$ being such that $\sum_{i \in I} c_i$ defined, and with $x_i \in X$ for $i \in I$.

Remark 3.7. As in the case of E_{T_S} , $P_{T_S}(P)$ can alternatively be defined as $\mu_1(T_S(P))$.

4. LINEAR TIME FIXPOINT LOGICS

We are now ready to define linear time fixpoint logics for coalgebras of type $T \circ F$, where $T : \text{Set} \rightarrow \text{Set}$ is the monad induced by a partial semiring $(S, +, 0, \bullet, 1)$, and $F : \text{Set} \rightarrow \text{Set}$ is a (typically polynomial) functor. Our logics will be valued into the semiring carrier S .

We fix a set Λ of modal operators with associated predicate liftings $(\llbracket \lambda \rrbracket)_{\lambda \in \Lambda}$ for F . A canonical choice for Λ and $(\llbracket \lambda \rrbracket)_{\lambda \in \Lambda}$ is given by Definition 3.2. Examples of non-canonical choices are considered later in this section.

Definition 4.1. Let \mathcal{V} be a set (of variables). The logic $\mu\mathcal{L}_\Lambda^\mathcal{V}$ has syntax given by

$$\mu\mathcal{L}_\Lambda^\mathcal{V} \ni \varphi ::= x \mid \llbracket \lambda \rrbracket(\varphi_1, \dots, \varphi_{\text{ar}(\lambda)}) \mid \sum_{i \in I} c_i \bullet \varphi_i \mid \mu x. \varphi \mid \nu x. \varphi$$

with $x \in \mathcal{V}$, $\lambda \in \Lambda$, I a finite set and $c_i \in S$ for $i \in I$ such that $\sum_{i \in I} c_i$ is defined.

For a $\mathsf{T}_S \circ F$ -coalgebra (C, γ) and a valuation $V : \mathcal{V} \rightarrow S^C$ (interpreting the variables in \mathcal{V} as S -valued predicates over C), the denotation $\llbracket \varphi \rrbracket_\gamma^V \in S^C$ of a formula $\varphi \in \mu\mathcal{L}_\Lambda^\mathcal{V}$ is defined inductively on the structure of φ by

- $\llbracket x \rrbracket_\gamma^V = V(x)$,
- $\llbracket \sum_{i \in I} (c_i \bullet \varphi_i) \rrbracket_\gamma^V = \mu_1(\sum_{i \in I} c_i \llbracket \varphi_i \rrbracket_\gamma^V)$,
- $\llbracket [\lambda](\varphi_1, \dots, \varphi_{\text{ar}(\lambda)}) \rrbracket_\gamma^V = \gamma^*(\text{ext}_{FC}(\llbracket \lambda \rrbracket_C(\llbracket \varphi_1 \rrbracket_\gamma^V, \dots, \llbracket \varphi_{\text{ar}(\lambda)} \rrbracket_\gamma^V)))$, where the *extension predicate lifting* $\text{ext} : S^\mathsf{T} \Rightarrow S^{\mathsf{T}S}$ is given by the action of E_{T_S} on objects, and where $\gamma^* : S^{\mathsf{T}SFC} \rightarrow S^C$ is reindexing along $\gamma : C \rightarrow \mathsf{T}_S FC$.
- $\llbracket \mu x. \varphi \rrbracket_\gamma^{V \setminus \{x\}}$ ($\llbracket \nu x. \varphi \rrbracket_\gamma^{V \setminus \{x\}}$) is the least (resp. greatest) fixpoint of the operator on S^C taking $p \in S^C$ to $\llbracket \varphi \rrbracket_\gamma^{V[p/x]}$, where the valuation $V[p/x] : \mathcal{V} \rightarrow S^C$ takes x to p and $y \in \mathcal{V} \setminus \{x\}$ to $V(y)$.

(In the second clause, both the formal sum notation and the action of the monad multiplication have been extended pointwisely to functions on C .) We write $\mu\mathcal{L}_\Lambda$ for the set of *closed* formulas ($\mathcal{V} = \emptyset$).

The fact that the operator used to interpret fixpoint formulas is order-preserving follows from the functoriality of predicate liftings. Existence of the required least and greatest fixpoints then follows by Theorem 2.7.

The semantics of $\mu\mathcal{L}_\Lambda$ resembles that of coalgebraic fixpoint logics (see e.g. [CKP11]), with two differences: (i) the interpretation of a formula is an *S-valued* predicate over the state space, and (ii) the extension lifting $E_{\mathsf{T}_S} : \text{Pred} \rightarrow \text{Pred}$ of Definition 3.6 is used to abstract away branching. In particular, the use of E_{T_S} is what makes $\mu\mathcal{L}_\Lambda$ a *linear time logic*.

Remark 4.2. Recall that our logics do not contain \top (for "true"). The formula $\nu x. \circ x$, with \circ the predicate lifting $\text{Pred}(F)$ of Remark 3.4, can be viewed as providing a *coinductive* interpretation of truth. For $S = (\{0, 1\}, \vee, 0, \wedge, 1)$, $\nu x. \circ x$ holds in a state precisely when there exists a maximal trace from that state, arising from a sequence of choices in the branching behaviour. (Such a path will not exist from a state that offers no choices for proceeding.) For $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, the truth value associated to $\nu x. \circ x$ in a particular state is the minimal cost that can be achieved along a maximal trace from that state. We write \top as a shorthand for the formula $\nu x. \circ x$, and conclude this remark by noting that its counterpart \perp (for "false") is already present in our logics as an empty weighted sum.

We also note that finite conjunctions are missing from the logics, and our step-wise semantics prevents their inclusion. (By this we mean that, with a step-wise semantics, it is not possible to include conjunction operators with the expected meaning, even in the case when $S = (\{0, 1\}, \vee, 0, \wedge, 1)$.) It is worth noting here that fixpoint logics for weighted systems are similar in their absence of conjunctions [Mei09]. For *total* semirings S , finite disjunctions are present as *1-weighted sums* (that is, finite weighted sums with all the weights equal to 1). Their interpretation is as expected: the formula $\sum_i 1 \bullet \varphi_i$ (written more simply $\sum_i \varphi_i$) measures the extent of conforming to one of the φ_i s. The next example shows how to include a form of disjunction in the logic also in the case of partial semirings S .

Example 4.3. Let $F = \coprod_{\lambda \in \Lambda} \text{Id}^{\text{ar}(\lambda)}$, with Λ a set of modal operators with specified arities. For each $\lambda, \lambda' \in \Lambda$ with $\lambda \neq \lambda'$, one can define a new modal operator $[\lambda]_- \sqcup [\lambda']_-$, of arity

equal to $\text{ar}(\lambda) + \text{ar}(\lambda')$, with associated predicate lifting $L : \text{Pred}^{\text{ar}(\lambda) + \text{ar}(\lambda')} \rightarrow \text{Pred}$ given by:

$$L_X(p_1, \dots, p_{\text{ar}(\lambda)}, p'_1, \dots, p'_{\text{ar}(\lambda')})(f) = \begin{cases} p_1(x_1) \bullet \dots \bullet p_{\text{ar}(\lambda)}(x_{\text{ar}(\lambda)}), & \text{if } f = \iota_\lambda(x_1, \dots, x_{\text{ar}(\lambda)}) \\ p'_1(x_1) \bullet \dots \bullet p'_{\text{ar}(\lambda')}(x_{\text{ar}(\lambda')}), & \text{if } f = \iota_{\lambda'}(x_1, \dots, x_{\text{ar}(\lambda')}) \\ 0, & \text{otherwise} \end{cases}.$$

Thus, the formula $[\lambda]\bar{\varphi} \sqcup [\lambda']\bar{\varphi}'$ incorporates a disjunction that can be resolved using a one-step unfolding of the coalgebra structure (as each "branch" resulting from a one-step unfolding will match either $[\lambda]$ or $[\lambda']$ or none of them). In the presence of such enhanced modalities, it is not too difficult to see that *deterministic* parity automata over F -structures² can be encoded as formulas of our logic that only use 1-weighted sums³.

Remark 4.4. The connection between our fixpoint logics and automata is outside the scope of this paper. An initial study of this connection [CSH] defines a notion of *parity $T_S \circ F$ -automaton* (as a $T_S \circ F$ -coalgebra equipped with a parity map), and shows that $\mu\mathcal{L}_\Lambda$ -formulas (assuming canonical predicate liftings for modal operators in Λ) can be translated to equivalent $T_S \circ F$ -automata. A translation in the opposite direction also exists, and is a direct generalisation of similar translations in the literature (see e.g. [Ven12]).

Example 4.5. To illustrate the use of modalities incorporating restricted disjunctions, let $F = A \times \text{Id}$ with A finite, and define $[\bar{a}]\varphi ::= \sqcup_{b \in A \setminus \{a\}} [b]\varphi$ for $a \in A$. Then, the extent to which $a \in A$ appears (i) eventually, (ii) always and (iii) infinitely often in the unfolding of a state in a $T \circ F$ -coalgebra is measured by the formulas $\mu x.([a]\top \sqcup [\bar{a}]x)$ (note here the use of \top as defined in Remark 4.2), $\nu x.[a]x$, and respectively $\nu x.\mu y.([a]x \sqcup [\bar{a}]y)$. Now take $F = 1 + A \times \text{Id}$ and let $[A]\varphi ::= \sqcup_{a \in A} [a]\varphi$. Then, the formula $\mu x.(\ast \sqcup [A]x)$ describes all finite words over A . All of the above formulas can be interpreted over coalgebras with either non-deterministic, probabilistic or weighted branching. Finally, when $S = ([0, 1], +, 0, \ast, 1)$ or $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, our logics can encode properties where weights are used to give preference to one observable outcome over another. For example, when $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, $F = A \times \text{Id}$ and $[\bar{a}]$ is as above, the formula $\nu x.\mu y.(0 \bullet [a]x + 1 \bullet [\bar{a}]y)$ introduces a penalty/cost whenever anything other than a is observed.

The next example compares the resulting logics with known logics for non-deterministic, probabilistic and weighted systems.

Examples 4.6. (1) For $S = (\{0, 1\}, \vee, 0, \wedge, 1)$, predicate liftings for F are as used in the semantics of coalgebraic modal logic [Pat03], and $\mu\mathcal{L}_\Lambda$ -formulas can be interpreted on F -coalgebras. In this case, assuming canonical choices for both Λ and $[\![\lambda]\!]$ with $\lambda \in \Lambda$ (see Definition 3.2), the logic $\mu\mathcal{L}_\Lambda$ can be viewed as an *existential* version of the linear time μ -calculus, wherein a linear time formula holds in a state whenever a trace satisfying the formula can be exhibited from that state. (To see that when $F = A \times \text{Id}$, the standard notion of linear time property coincides with ours, note that the former is equivalent to a non-deterministic Büchi automaton, which can, in turn, be captured by a fixpoint formula in our logic – the latter does not require any use of conjunction operators!) Our logics are, however, more general, as they apply to transition structures defined by arbitrary polynomial endofunctors F , and to arbitrary choices for the set Λ of predicate liftings.

²This includes infinite words ($F = A \times \text{Id}$) and infinite trees ($F = A \times \text{Id} \times \text{Id}$).

³For partial semirings S , the only 1-weighted sums are the trivial ones.

- (2) For $S = ([0, 1], +, 0, *, 1)$, $\mu\mathcal{L}_\Lambda$ -formulas measure the likelihood of satisfying a certain linear time property. This time, a linear time property in our setting differs somewhat from the standard notion, as our logics use sub-convex combinations of formulas instead of arbitrary disjunctions, and no conjunctions. However, under the assumption that F specifies *word-like behaviour* (with associated 0-ary and unary modalities only), as used in both LTL and the linear time μ -calculus, one can encode both LTL and linear time μ -calculus formulas into our logics by translating first to deterministic parity automata over F -structures, and subsequently making use of modalities incorporating disjunctions (see Example 4.3) to encode such deterministic parity automata into our logics. Note that, for word-like behaviours, deterministic and non-deterministic parity automata are equally expressive [GTW02].
- (3) For $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$ (and its bounded variant), $\mu\mathcal{L}_\Lambda$ -formulas measure the minimal cost of satisfying a certain linear time property. While we are not aware of any temporal logics on weighted transition systems with a semantics similar to ours, the notion of *weighted Büchi automaton* [DR06], parameterised by a *totally complete semiring*, is an instance of the $\mathsf{T}_S \circ F$ -automata of [CSH], and therefore subsumed by our logics.

5. PATH-BASED SEMANTICS FOR QUANTITATIVE LINEAR TIME LOGICS

This section provides an alternative path-based semantics for the *fixpoint-free fragment* of $\mu\mathcal{L}_\Lambda^\vee$, and proves its equivalence to the already-defined step-wise semantics. This is done by (i) rephrasing the step-wise semantics of the fixpoint-free fragment in categorical terms, using dual adjunctions, (ii) defining a path-based semantics for the fixpoint-free fragment, also in categorical terms, and (iii) proving the equivalence of the two semantics, first for the *modal* fragment of $\mu\mathcal{L}_\Lambda^\vee$ (no propositional or fixpoint operators), and then for the fixpoint-free fragment of $\mu\mathcal{L}_\Lambda^\vee$ (with propositional operators now allowed in formulas). The main results (Theorem 5.15 and Theorem 5.23) assume canonical choices for both the branching and the linear time modalities, but a generalisation to non-canonical choices, subject to additional requirements, is also discussed (Theorem 5.17).

Giving an alternative path-based semantics for the *full* language $\mu\mathcal{L}_\Lambda^\vee$ is difficult at the level of generality considered here. On the one hand, for probabilistic systems ($S = ([0, 1], +, 0, *, 1)$), a path-based semantics in the case when F specifies word-like behaviour requires measure-theoretic concepts (see e.g. [BK08, Chapter 10]). On the other hand, for weighted systems (where we are not aware of similar logics), defining the behaviour of a weighted Büchi automaton involves taking infinite sums over all successful paths of the automaton over an individual word. This requires the existence of both infinite sums and infinite products in the underlying semiring [DR06]. It is not clear how these two different path-based approaches can be unified. However, following recent results on a close connection between our canonical logics and quantitative parity automata [CSH], one can argue that a path-based semantics for the full fixpoint language is not needed, as the automata-theoretic semantics described in loc. cit. provides sufficient evidence that our logics coincide with known linear time logics for specific model types.

5.1. Coalgebraic Linear Time Logics via Dual Adjunctions. This section rephrases the predicate lifting approach to defining the semantics of coalgebraic linear time logics in

terms of dual adjunctions. Such a categorical presentation eases the proof of equivalence with the (yet to be defined) path-based semantics.

For an endofunctor $F : \mathbf{Set} \rightarrow \mathbf{Set}$, the dual adjunction approach to defining a logic for

F -coalgebras involves a contravariant adjunction $\mathcal{A} \begin{smallmatrix} \xrightarrow{Sp} \\ \perp \\ \xleftarrow{P} \end{smallmatrix} \mathbf{Set}^{\text{op}}$, a functor $L : \mathcal{A} \rightarrow \mathcal{A}$ and

a natural transformation $\delta : LP \Rightarrow PF$. Given these ingredients and a set \mathcal{V} of variables, the free L -algebra $(\mathcal{L}^{\mathcal{V}}, \alpha^{\mathcal{V}})$ over \mathcal{V} provides a language for F -coalgebras, with formulas built from variables in \mathcal{V} and modal operators as specified by L . Given an F -coalgebra (X, γ) and a valuation $V : \mathcal{V} \rightarrow PX$, the associated semantics $\llbracket _ \rrbracket_{\gamma}^V : \mathcal{L}^{\mathcal{V}} \rightarrow PX$ is the unique L -algebra homomorphism from $\alpha^{\mathcal{V}}$ to $P\gamma \circ \delta_X$ which extends V :

$$\begin{array}{ccc} L(\mathcal{L}^{\mathcal{V}}) & \xrightarrow{L\llbracket _ \rrbracket_{\gamma}^V} & LPX \\ \alpha^{\mathcal{V}} \downarrow & & \downarrow \delta_X \\ \mathcal{L}^{\mathcal{V}} & \xrightarrow{\llbracket _ \rrbracket_{\gamma}^V} & PX \\ & & \downarrow P\gamma \\ & & PF\gamma \end{array}$$

To match the syntax and semantics of the *modal fragment* of our logic $\mu\mathcal{L}_{\Lambda}^{\mathcal{V}}$ (no propo-

sitional or fixpoint operators), we consider the dual adjunction $\mathbf{Set} \begin{smallmatrix} \xrightarrow{Sp} \\ \perp \\ \xleftarrow{P} \end{smallmatrix} \mathbf{Set}^{\text{op}}$ with

$Sp = P = S^{\top}$. The natural transformation of type $LP \Rightarrow PTF$ required by such an approach is obtained by combining *two* natural transformations, of type $LP \Rightarrow PF$ and $P \Rightarrow PT$, respectively. These, in turn, are determined by the choice of predicate liftings $\llbracket \lambda \rrbracket$ for F , and by the use of the canonical extension lifting ext , respectively. Specifically:

- (1) $L : \mathbf{Set} \rightarrow \mathbf{Set}$ is given by $LX := \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}$, and $\delta : LP \Rightarrow PF$ is given by

$$\delta_X(\iota_{\lambda}(p_1, \dots, p_{\text{ar}(\lambda)})) = \llbracket \lambda \rrbracket(p_1, \dots, p_{\text{ar}(\lambda)})$$

for $\lambda \in \Lambda$ and $p_1, \dots, p_{\text{ar}(\lambda)} \in PX$.

- (2) $\sigma : P = \text{Id}P \Rightarrow PT$ is given by

$$\sigma_X(p) = \mu_1 \circ \top p \tag{5.1}$$

for $p \in PX$ (see also Remark 3.7). The use of the identity functor to define a syntax for

\top reflects the fact that, in our logics, the branching modality is hidden from the syntax.

The modal fragment of the logic $\mu\mathcal{L}_{\Lambda}^{\mathcal{V}}$ from Section 4 then coincides with the logic $\mathcal{L}^{\mathcal{V}}$ induced by the natural transformation $\sigma_F \circ \delta$:

$$LP \xRightarrow{\delta} PF \xRightarrow{\sigma_F} PTF$$

That is, for a $\top F$ -coalgebra (X, γ) and a valuation $V : \mathcal{V} \rightarrow PX$, the map $\llbracket _ \rrbracket_{\gamma}^V : \mathcal{L}_{\Lambda}^{\mathcal{V}} \rightarrow PX$ is defined as the unique L -algebra homomorphism from the free L -algebra $(\mathcal{L}^{\mathcal{V}}, \alpha^{\mathcal{V}})$ over \mathcal{V} to $(PX, P\gamma \circ \sigma_{FX} \circ \delta_X)$ which extends V .

Other choices for a modality that abstracts away branching have been considered in related work: both [Has15] and [KR15] propose using an arbitrary \top -algebra structure $\tau : \top^2 1 \rightarrow \top 1$ instead of μ_1 in the definition of σ . While most of the results in this paper concern the canonical choice for σ (which we believe is precisely what justifies our use of

the term "linear time"), we will also explore the more general σ s arising from a choice of τ as above. For this, we will need the following lemma.

Lemma 5.1. *For any T -algebra $(\mathsf{T}1, \tau)$, with induced $\sigma : P \Rightarrow P\mathsf{T}$, we have $\sigma_{\mathsf{T}} \circ \sigma = P\mu \circ \sigma$.*

Proof. $\sigma_{\mathsf{T}X} \circ \sigma_X$ maps a predicate $p : X \rightarrow P1$ to the predicate $\tau \circ \mathsf{T}\tau \circ \mathsf{T}^2p$, whereas $P\mu_X \circ \sigma_X$ maps p to $\tau \circ \mathsf{T}p \circ \mu_X$. The conclusion now follows from the commutativity of

$$\begin{array}{ccccc} \mathsf{T}^2X & \xrightarrow{\mathsf{T}^2p} & \mathsf{T}^31 & \xrightarrow{\mathsf{T}\tau} & \mathsf{T}^21 \\ \mu_X \downarrow & & \mu_{\mathsf{T}1} \downarrow & & \downarrow \tau \\ \mathsf{T}X & \xrightarrow{\mathsf{T}p} & \mathsf{T}^21 & \xrightarrow{\tau} & \mathsf{T}1 \end{array}$$

where the left and right squares follow by naturality of μ and by τ being a T -algebra, respectively. \square

The next definition describes the canonical choice for the corresponding L and $\delta : LP \Rightarrow PF$, arising from the canonical choice for Λ and $(\llbracket \lambda \rrbracket)_{\lambda \in \Lambda}$ (as given in Definition 3.2).

Definition 5.2. Let $L ::= F = \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}$, with Λ a set of modal operators with specified arities. Also, let $\delta : LP \Rightarrow PF$ be given by

$$\begin{array}{ccc} (PX)^{\text{ar}(\lambda)} & \xrightarrow{\bullet_X \circ (P\pi_1 \times \dots \times P\pi_{\text{ar}(\lambda)})} & P(X^{\text{ar}(\lambda)}) \xrightarrow{e_\lambda} P(\coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}) \\ \downarrow \iota_\lambda & \dashrightarrow & \delta_X \\ \coprod_{\lambda \in \Lambda} (PX)^{\text{ar}(\lambda)} & \dashrightarrow & \end{array}$$

where in the above $\bullet_Y : (PY)^n \rightarrow PY$ is given by the transpose of the map

$$(S^Y \times \dots \times S^Y) \times Y \longrightarrow S, \quad (p_1, \dots, p_n, y) \mapsto p_1(y) \bullet \dots \bullet p_n(y)$$

with $\bullet : S \times S \rightarrow S$ the multiplication operation on S (extended to an n -ary operation), and with $e_\lambda : P(X^{\text{ar}(\lambda)}) \rightarrow P(\coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)})$ being given by

$$X^{\text{ar}(\lambda)} \xrightarrow{p} S \xrightarrow{e_\lambda} \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)} \xrightarrow{[0, \dots, p, \dots, 0]} S$$

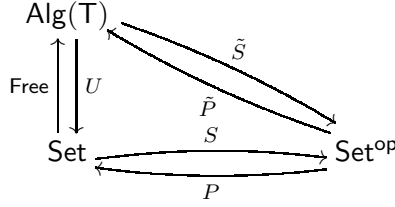
We proceed by showing how to recover the whole fixpoint-free fragment of $\mu\mathcal{L}_\Lambda^\vee$ (including propositional operators). This requires moving to the category $\text{Alg}(\mathsf{T})$ of Eilenberg-Moore algebras of T .

It follows e.g. from [Jac16, Exercise 5.4.11] that for T a strong monad and $(A, \alpha) \in$

$$\text{Alg}(\mathsf{T}), \text{ the dual adjunction } \text{Set} \begin{array}{c} \xrightarrow{S} \\ \perp \\ \xleftarrow{P} \end{array} \text{Set}^{\text{op}} \text{ with } S = P = A^\top \text{ lifts to } \text{Alg}(\mathsf{T}) \begin{array}{c} \xrightarrow{\tilde{S}} \\ \perp \\ \xleftarrow{\tilde{P}} \end{array} \text{Set}^{\text{op}},$$

with $\tilde{S} = (A, \alpha)^\top$ and $\tilde{P} = A^\top$, where the T -algebra required in the definition of $\tilde{P}X$ is the transpose of $\mathsf{T}(A^X) \times X \xrightarrow{\text{st}'_{A^X, X}} \mathsf{T}(A^X \times X) \xrightarrow{\mathsf{T}\text{eval}} \mathsf{T}A \xrightarrow{\alpha} A$. We then have $S = \tilde{S}\text{Free}$ and $P = U\tilde{P}$, where $\text{Free} : \text{Set} \rightarrow \text{Alg}(\mathsf{T})$ takes X to $(\mathsf{T}X, \mu_X)$ and $U : \text{Alg}(\mathsf{T}) \rightarrow \text{Set}$ takes

(B, β) to B :



As before, our choice of (A, α) will be either $(T1, \mu_1)$ or an arbitrary $(T1, \tau) \in \text{Alg}(T)$. Irrespective of this, we can lift the functor $L : \text{Set} \rightarrow \text{Set}$ to $\tilde{L} : \text{Alg}(T) \rightarrow \text{Alg}(T)$ by taking $\tilde{L} = \text{Free}LU$. Then, the one-step semantics $\delta : LU\tilde{P} = LP \Rightarrow PF = U\tilde{P}F$ lifts to $\tilde{\delta} ::= \delta^\# : \tilde{L}\tilde{P} = \text{Free}LU\tilde{P} \Rightarrow \tilde{P}F$.

There is no need for a similar lifting of the identity functor on Set with associated one-step semantics σ to $\text{Alg}(T)$, since the components of σ are already $\text{Alg}(T)$ -homomorphisms – this follows from an equivalent definition of $\sigma_X : (T1)^X \rightarrow (T1)^{TX}$ as the transpose of the unique extension of $\text{eval} : (T1)^X \times X \rightarrow T1$ to a 2-linear map⁴ $(T1)^X \times TX \rightarrow T1$, as shown in [Koc12, Proposition 4.1]. We therefore simply write $\tilde{\sigma} : \tilde{P} \Rightarrow \tilde{P}T$ for the natural transformation whose components are given by those of $\sigma : P \Rightarrow PT$.

This yields a new logic $\tilde{\mathcal{L}}^{\text{Free}(\mathcal{V})}$ carrying T -algebra structure, with associated semantics $\llbracket _ \rrbracket_\gamma^{V^\#} : \tilde{\mathcal{L}}^{\text{Free}(\mathcal{V})} \rightarrow \tilde{P}X$, for each TF -coalgebra (X, γ) and valuation $V : \mathcal{V} \rightarrow PX$ (extending to a T -algebra homomorphism $V^\# : \text{Free}(\mathcal{V}) \rightarrow \tilde{P}X$). The syntax of the resulting logic contains propositional operators arising from the T -algebra structure. We have therefore recovered precisely the fixpoint-free fragment of the logic $\mu\mathcal{L}_\Lambda^\mathcal{V}$ from Section 4, as further illustrated by the next example.

- Examples 5.3.** (1) For $S = (\{0, 1\}, \vee, 0, \wedge, 1)$, $\text{Alg}(T)$ is isomorphic to the category of join semi-lattices, and the enhanced logic contains finite disjunctions (equivalent to finite sums $\sum_{i \in I} c_i \varphi_i$ with I finite and $c_i \in \{0, 1\}$ for $i \in I$).
- (2) For $S = ([0, 1], +, 0, *, 1)$, $\text{Alg}(T)$ is isomorphic to the category of positive convex algebras, and the enhanced logic contains sub-convex combinations of formulas.
- (3) For $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$ or $S = ([0, B]^\infty, \min, \infty, +_B, 0)$ with $B \in \mathbb{N}$, $\text{Alg}(T)$ is isomorphic to the category of modules over S , and the enhanced logic contains finite linear combinations of formulas.

5.2. Path-Based Semantics for $\mathcal{L}^\mathcal{V}$. We first consider the modal fragment of $\mu\mathcal{L}_\Lambda^\mathcal{V}$, which coincides with the logic $\mathcal{L}^\mathcal{V}$ of Section 5.1. We define a path-based semantics for this fragment, and prove its equivalence to the step-wise semantics.

For a polynomial endofunctor F and a commutative monad T , one can define a canonical distributive law of T over F as shown below. This will be used to give a path-based semantics for the modal fragment of $\mathcal{L}_\Lambda^\mathcal{V}$, by delaying the use of the natural transformation σ when defining the interpretation of modal formulas in $\mathcal{L}_\Lambda^\mathcal{V}$ for as long as possible.

⁴A 2-linear map is required to preserve the T -algebra structure in the second argument, where the assumed T -algebra structures on TX and $T1$ are the free ones (μ_X and μ_1 respectively).

Definition 5.4. For $F = \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}$ the *canonical distributive law* $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ is given by

$$\begin{array}{ccc} (\mathbb{T}X)^{\text{ar}(\lambda)} & \xrightarrow{\text{dst}_{\text{ar}(\lambda)}} & \mathbb{T}(X^{\text{ar}(\lambda)}) \xrightarrow{\mathbb{T}\iota_\lambda} \mathbb{T}(\coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}) = \mathbb{T}FX \\ \downarrow \iota_\lambda & \dashrightarrow \lambda_X & \\ F\mathbb{T}X = \coprod_{\lambda \in \Lambda} (\mathbb{T}X)^{\text{ar}(\lambda)} & & \end{array}$$

where $\text{dst}_n : (\mathbb{T}X)^n \rightarrow \mathbb{T}(X^n)$ is either $\eta_1 : 1 \rightarrow \mathbb{T}1$ (if $n = 0$), the identity map (if $n = 1$), or defined in the obvious way from the double strength of the monad \mathbb{T} (if $n \geq 2$).

It follows directly from the above definition that the distributive law λ is compatible with the monad structure, that is:

$$\begin{array}{ccc} FX & \xrightarrow{F\eta_X} & F\mathbb{T}X \\ \searrow \eta_{FX} & & \downarrow \lambda_X \\ & & \mathbb{T}FX \end{array} \quad \begin{array}{ccccc} F\mathbb{T}^2X & \xrightarrow{\lambda_{\mathbb{T}X}} & \mathbb{T}F\mathbb{T}X & \xrightarrow{\mathbb{T}\lambda_X} & \mathbb{T}^2FX \\ F\mu_X \downarrow & & & & \downarrow \mu_{FX} \\ F\mathbb{T}X & \xrightarrow{\lambda_X} & \mathbb{T}FX & & \end{array}$$

Example 5.5. In the particular case when $\mathbb{T} = \mathbb{T}_S$, the canonical distributive law $\lambda : F\mathbb{T}_S \Rightarrow \mathbb{T}_S F$ takes $\iota_\lambda(f_1, \dots, f_{\text{ar}(\lambda)})$ with $f_1, \dots, f_{\text{ar}(\lambda)} \in \mathbb{T}_S X = S^X$ to $f \in \mathbb{T}_S FX = S^{FX}$ given by

$$\begin{aligned} f(\iota_\lambda(x_1, \dots, x_{\text{ar}(\lambda)})) &= f_1(x_1) \bullet \dots \bullet f_{\text{ar}(\lambda)}(x_{\text{ar}(\lambda)}), \\ f(\iota_{\lambda'}(x_1, \dots, x_{\text{ar}(\lambda')})) &= 0, \text{ for } \lambda' \neq \lambda. \end{aligned}$$

Unfolding the coalgebra map $\gamma : X \rightarrow \mathbb{T}FX$ $n \geq 1$ times (as required in order to interpret formulas of modal depth n) yields a map $(\mathbb{T}F)^{n-1}\gamma \circ \dots \circ \gamma : X \rightarrow (\mathbb{T}F)^n X$. Alternatively, one can use the distributive law λ to flatten, at each unfolding step, the branching arising from the presence of \mathbb{T} in the coalgebra type:

Definition 5.6. For a natural transformation $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ and a $\mathbb{T}F$ -coalgebra (X, γ) , let $\gamma_n : X \rightarrow \mathbb{T}F^n X$ with $n \geq 1$ be given by

- $\gamma_1 = \gamma$
- $\gamma_{n+1} = \mu_{F^{n+1}X} \circ \mathbb{T}\lambda_{F^n X} \circ \mathbb{T}F\gamma_n \circ \gamma$:

$$X \xrightarrow{\gamma} \mathbb{T}FX \xrightarrow{\mathbb{T}F\gamma_n} \mathbb{T}F\mathbb{T}F^n X \xrightarrow{\mathbb{T}\lambda_{F^n X}} \mathbb{T}^2 F^{n+1} X \xrightarrow{\mu_{F^{n+1}X}} \mathbb{T}F^{n+1} X$$

In order to relate the maps $(\mathbb{T}F)^{n-1}\gamma \circ \dots \circ \gamma$ and γ_n , with $n \geq 1$, we first note that any distributive law $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ yields natural transformations $(\mathbb{T}F)^n \Rightarrow \mathbb{T}F^n$:

Definition 5.7. For a natural transformation $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$, let $\lambda_n : (\mathbb{T}F)^n \Rightarrow \mathbb{T}F^n$ for $n \geq 0$ be defined inductively by:

- $\lambda_0 = \eta : \text{Id} \rightarrow \mathbb{T}$
- $\lambda_{n+1} = \mu_{F^{n+1}} \circ \mathbb{T}\lambda_{F^n} \circ \mathbb{T}F\lambda_n$ for $n \geq 1$:

$$\mathbb{T}F(\mathbb{T}F)^n \xrightarrow{\mathbb{T}F\lambda_n} \mathbb{T}F\mathbb{T}F^n \xrightarrow{\mathbb{T}\lambda_{F^n}} \mathbb{T}^2 F^{n+1} \xrightarrow{\mu_{F^{n+1}}} \mathbb{T}F^{n+1}$$

In particular, Definition 5.7 yields $\lambda_1 = \mu_F \circ \mathbb{T}\lambda \circ \mathbb{T}F\eta = \mu_F \circ \mathbb{T}\eta = \text{id}$ (where the second equality follows by the compatibility of λ with the monad structure).

We can now state the following:

Lemma 5.8. *Let (X, γ) be a TF -coalgebra and $\lambda : \mathsf{TF} \Rightarrow F\mathsf{T}$ be a distributive law of T over F . For $n \geq 1$, we have $\gamma_n = \lambda_n \circ (\mathsf{TF})^{n-1}\gamma \circ \dots \circ \gamma$.*

Proof. Induction on n . The base case is trivial. The inductive step follows from the commutativity of

$$\begin{array}{ccccccc} X & \xrightarrow{\gamma} & \mathsf{TF}X & \xrightarrow{\mathsf{TF}\gamma_n} & \mathsf{TF}\mathsf{TF}^n X & \xrightarrow{\mathsf{T}\lambda_{F^n X}} & \mathsf{T}^2 F^{n+1} X \xrightarrow{\mu_{F^{n+1}}} \mathsf{TF}^{n+1} X \\ \downarrow \gamma & & & & \uparrow \mathsf{TF}\lambda_n & \nearrow \lambda_{n+1} & \\ \mathsf{TF}X & \xrightarrow{\mathsf{TF}((\mathsf{TF})^{n-1}\gamma \circ \dots \circ \gamma)} & (\mathsf{TF})^{n+1} X & & & & \end{array}$$

which, in turn, follows by the induction hypothesis and the definition of λ_{n+1} . \square

In order to define an alternative, path-based semantics for $\mathcal{L}^\mathcal{V}$, we let $F_X : \mathsf{Set} \rightarrow \mathsf{Set}$ be given by $F_X Y = X \times FY$. Then a TF -coalgebra (X, γ) yields a TF_X -coalgebra $(X, \mathsf{st}_{X, FX} \circ \langle \mathsf{id}_X, \gamma \rangle)$:

$$X \xrightarrow{\langle \mathsf{id}_X, \gamma \rangle} X \times \mathsf{TF}X \xrightarrow{\mathsf{st}_{X, FX}} \mathsf{TF}_X X.$$

The intention is to interpret formulas of modal depth n over variables in \mathcal{V} via an intermediate interpretation over *paths of depth n* , i.e. elements of $F_X^n X$. (The use of F_X is required to deal with variables.) To this end, we let $\mathcal{L}_0^\mathcal{V} = \mathcal{V}$ and $\mathcal{L}_{n+1}^\mathcal{V} = \mathcal{V} + L\mathcal{L}_n^\mathcal{V}$ for $n \in \omega$, and note that $\mathcal{L}^\mathcal{V} = \bigcup_{n \in \omega} \mathcal{L}_n^\mathcal{V}$. We also note that the natural transformation $\lambda : F\mathsf{T} \Rightarrow \mathsf{TF}$ extends to a natural transformation $\lambda_X : F_X \mathsf{T} \Rightarrow \mathsf{TF}_X$ whose components are given by:

$$F_X \mathsf{T}Y = X \times F\mathsf{T}Y \xrightarrow{\mathsf{id}_X \times \lambda_Y} X \times \mathsf{TF}Y \xrightarrow{\mathsf{st}_{X, FY}} \mathsf{T}(X \times FY) = \mathsf{TF}_X Y$$

Now we can instantiate Definitions 5.6 and 5.7 with $\lambda_X : F_X \mathsf{T} \Rightarrow \mathsf{TF}_X$ and $\gamma_X : X \rightarrow \mathsf{TF}_X X$ to obtain maps $(\gamma_X)_n : X \rightarrow \mathsf{TF}_X^n X$ and $(\lambda_X)_n : (\mathsf{TF}_X)^n \Rightarrow \mathsf{T}(F_X)^n$ for $n \geq 1$. In particular, Lemma 5.8 holds for γ_X and λ_X .

Definition 5.9 (Path-Based Semantics for $\mathcal{L}^\mathcal{V}$). Let (X, γ) be a TF -coalgebra, and let $V : \mathcal{V} \rightarrow PX$ be a valuation. Let $\rho_n : \mathcal{L}_n^\mathcal{V} \rightarrow PF_X^n X$ with $n \in \omega$ be given by:

- $\rho_0 = V : \mathcal{L}_0^\mathcal{V} \rightarrow PX$,
- $\rho_{n+1} = [P\pi_1 \circ V, (\delta_X)_{F_X^n X} \circ L\rho_n] : \mathcal{L}_{n+1}^\mathcal{V} \rightarrow PF_X^{n+1} X$ for $n \geq 0$:

$$\mathcal{V} \xrightarrow{V} PX \xrightarrow{P\pi_1} PF_X^{n+1} X \xleftarrow{(\delta_X)_{F_X^n X}} LPF_X^n X \xleftarrow{L\rho_n} L\mathcal{L}_n^\mathcal{V}$$

where we write $\delta_X : LP \Rightarrow PF_X$ for the natural transformation given by $P\pi_2 \circ \delta$. Finally, for $\varphi \in \mathcal{L}_n^\mathcal{V}$, the *path-based semantics* $\llbracket \varphi \rrbracket_\gamma^V$ is given by the image of φ under the map

$$\mathcal{L}_n^\mathcal{V} \xrightarrow{\rho_n} PF_X^n X \xrightarrow{\sigma_{F_X^n X}} P\mathsf{TF}_X^n X \xrightarrow{P(\gamma_X)_n} PX$$

Thus, in the path-based semantics, in order to interpret a formula of modal depth n , the n -step behaviour of a state in a TF_X -coalgebra is flattened into branches of n -step F_X -behaviours (using $(\gamma_X)_n$), and this results in the natural transformation σ (or equivalently, the extension lifting ext) only being used once, rather than at each unfolding of the coalgebra map.

Our goal now is to show that the step-wise and path-based semantics for the modal fragment of $\mathcal{L}^\mathcal{V}$ are equivalent. For this, we need the following inductive formulation of the step-wise semantics:

Definition 5.10. Let $V : \mathcal{V} \rightarrow PX$ be a valuation. For $n \geq 1$, let $\xi_n : \mathcal{L}_n^\mathcal{V} \rightarrow P(\mathsf{T}F_X)^n X$ be defined by:

- $\xi_0 = V : \mathcal{L}_0^\mathcal{V} \rightarrow PX$,
- $\xi_{n+1} = \sigma_{F_X(\mathsf{T}F_X)^n X} \circ [P\pi_1 \circ V, (\delta_X)_{(\mathsf{T}F_X)^n X} \circ L\xi_n]$ for $n \geq 1$:

$$\mathcal{L}_{n+1}^\mathcal{V} = \mathcal{V} + L\mathcal{L}_n^\mathcal{V} \xrightarrow{[P\pi_1 \circ V, (\delta_X)_{(\mathsf{T}F_X)^n X} \circ L\xi_n]} PF_X(\mathsf{T}F_X)^n X \xrightarrow{\sigma_{F_X(\mathsf{T}F_X)^n X}} P(\mathsf{T}F_X)^{n+1} X$$

Lemma 5.11. For formulas in $\mathcal{L}_n^\mathcal{V}$, the step-wise semantics is obtained by post-composing ξ_n with $P\gamma_X \circ \dots \circ P(\mathsf{T}F_X)^{n-1}\gamma_X : P(\mathsf{T}F_X)^n X \rightarrow PX$.

Proof. Immediate. \square

The last ingredient required for the proof of equivalence of the two semantics is the following key lemma, which allows us to move from alternating the use of the natural transformations σ and δ (as done in the step-wise semantics) to only using the natural transformation σ once (as done in the path-based semantics).

Since later in the paper we discuss other choices for σ , obtained by replacing $(\mathsf{T}1, \mu_1)$ with an arbitrary T -algebra $(\mathsf{T}1, \tau)$, most of the proof of the lemma uses τ instead of μ_1 .

Lemma 5.12. Let $\delta : LP \Rightarrow PF$ and $\lambda : F\mathsf{T} \Rightarrow \mathsf{T}F$ be as in Definitions 5.2 and 5.4, respectively, and let $\sigma : P \Rightarrow P\mathsf{T}$ be the natural transformation induced by $\tau := \mu_1 : \mathsf{T}^2 1 \rightarrow \mathsf{T}1$, given by (5.1). Then the following diagram commutes:

$$\begin{array}{ccc} LP & \xrightarrow{L\sigma} & LPT \xrightarrow{\delta_{\mathsf{T}}} PFT \\ \delta \Downarrow & & \Uparrow P\lambda \\ PF & \xrightarrow{\sigma_F} & P\mathsf{T}F \end{array} \quad (5.2)$$

Proof. The statement follows by expanding the corresponding definitions of δ and σ :

Given $(p_i) \in \iota_\lambda(PX)^n$ (with $n = \text{ar}(\lambda) \geq 2$), we have:

$$\left(\begin{array}{c} X \\ \downarrow p_i \\ \mathsf{T}1 \end{array} \right) \xrightarrow{L\sigma_X} \left(\begin{array}{c} \mathsf{T}X \\ \downarrow \mathsf{T}p_i \\ \mathsf{T}^2 1 \\ \downarrow \tau \\ \mathsf{T}1 \end{array} \right) \xrightarrow{\delta_{\mathsf{T}X}} \coprod_{\lambda \in \Lambda} (\mathsf{T}X)^{\text{ar}(\lambda)} \downarrow_{[0, \dots, \bullet(\tau \circ \mathsf{T}p_i), \dots, 0]} \mathsf{T}1$$

and

$$\begin{array}{ccccccc} & & & & & \coprod_{\lambda \in \Lambda} (\mathsf{T}X^{\text{ar}(\lambda)}) & \\ & & & & & \downarrow +_{\lambda \in \Lambda} \text{dst}_{\text{ar}(\lambda)} & \\ & & & & & \coprod_{\lambda \in \Lambda} \mathsf{T}(X^{\text{ar}(\lambda)}) & \\ & & & & & \downarrow [\mathsf{T}t_1, \dots, \mathsf{T}t_n] & \\ & & & & & \mathsf{T} \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)} & \\ & & & & & \downarrow \mathsf{T}[0, \dots, \bullet p_i, \dots, 0] & \\ & & & & & \mathsf{T}^2 1 & \\ & & & & & \downarrow \tau & \\ & & & & & \mathsf{T}1 & \\ \left(\begin{array}{c} X \\ \downarrow p_i \\ \mathsf{T}1 \end{array} \right) & \xrightarrow{\delta_X} & \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)} & \downarrow_{[0, \dots, \bullet p_i, \dots, 0]} \mathsf{T}1 & \xrightarrow{\sigma_{FX}} & \mathsf{T} \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)} & \downarrow_{\mathsf{T}[0, \dots, \bullet p_i, \dots, 0]} \mathsf{T}^2 1 \downarrow \tau \mathsf{T}1 \\ & & & & & \downarrow \mathsf{T}[0, \dots, \bullet p_i, \dots, 0] & \\ & & & & & \mathsf{T}^2 1 & \\ & & & & & \downarrow \tau & \\ & & & & & \mathsf{T}1 & \end{array}$$

where for $q_i : X \rightarrow \mathbb{T}1$ with $i \in \{1, \dots, n\}$, $\bullet(p_i) : X^n \rightarrow \mathbb{T}1$ takes (x_1, \dots, x_n) to $p_1(x_1) \bullet \dots \bullet p_n(x_n)$. Thus, the commutativity of (5.2) amounts to the commutativity of

$$\begin{array}{ccccc} \mathbb{T}X \times \mathbb{T}X & \xrightarrow{\mathbb{T}p_1 \times \mathbb{T}p_2} & \mathbb{T}^2 1 \times \mathbb{T}^2 1 & \xrightarrow{\tau \times \tau} & \mathbb{T}1 \times \mathbb{T}1 \\ \text{dst}_{X,X} \downarrow & & \text{dst}_{\mathbb{T}1, \mathbb{T}1} \downarrow & & \downarrow \bullet \\ \mathbb{T}(X \times X) & \xrightarrow{\mathbb{T}(p_1 \times p_2)} & \mathbb{T}(\mathbb{T}1 \times \mathbb{T}1) & \xrightarrow{\mathbb{T}\bullet} \mathbb{T}^2 1 & \xrightarrow{\tau} \mathbb{T}1 \end{array} \quad (5.3)$$

where for simplicity we assume $n = 2$. For $\tau = \mu_1$, the latter follows easily by naturality of dst (left rectangle) and exploiting the equivalent definition of \bullet as $\mathbb{T}\pi_1 \circ \text{dst}_{\mathbb{T}1, \mathbb{T}1}$, as given in [C r17] (right rectangle). The proof in the case when $\text{ar}(\lambda) = 1$ is trivial, whereas the proof in the case when $\text{ar}(\lambda) = 0$ uses the fact that $(\mathbb{T}1, \tau)$ is a \mathbb{T} -algebra (and hence $\tau \circ \mathbb{T}\eta_1 = \text{id}$). \square

Remark 5.13. The commutativity of (5.3) in the proof of Lemma 5.12 relies on the well-behavedness of μ_1 w.r.t. the double strength map. Replacing $\mu_1 : \mathbb{T}^2 1 \rightarrow \mathbb{T}1$ by an arbitrary \mathbb{T} -algebra structure $\tau : \mathbb{T}^2 1 \rightarrow \mathbb{T}1$ will not, in general, make this diagram commute. When \mathbb{T} is the finite powerset functor ($S = (\{0, 1\}, \vee, 0, \wedge, 1)$), an example is the \Box -modality $\tau_\Box : \mathbb{T}^2 1 \rightarrow \mathbb{T}1$, defined from the \Diamond -modality μ_1 via the swap map $\text{swap} : \mathbb{T}1 \rightarrow \mathbb{T}1$: $\tau = \text{swap} \circ \mu_1 \circ \mathbb{T}\text{swap}$; an easy calculation shows that commutativity of the previously mentioned diagram fails in this case.

Corollary 5.14. *Let δ, λ and σ be as in Lemma 5.12, and let $\delta_X : LP \Rightarrow PF_X$ be as in Definition 5.9. Then the following diagram commutes:*

$$\begin{array}{ccc} LP & \xrightarrow{L\sigma} LPT \xrightarrow{(\delta_X)_\mathbb{T}} PF_X \mathbb{T} \\ \delta_X \Downarrow & & \Uparrow P\lambda_X \\ PF_X & \xrightarrow{\sigma_{F_X}} P\mathbb{T}F_X \end{array} \quad (5.4)$$

Proof. Immediate from Lemma 5.12 and the definitions of δ_X and λ_X . \square

Using Corollary 5.14, we can now state and prove the announced equivalence result.

Theorem 5.15. *Let $\delta : LP \Rightarrow PF$, $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ and $\sigma : P \Rightarrow P\mathbb{T}$ be as in Lemma 5.12. Also, let (X, γ) be a $\mathbb{T}F$ -coalgebra and let $V : \mathcal{V} \rightarrow PX$ be a valuation. For $\varphi \in \mathcal{L}^\mathcal{V}$, $\llbracket \varphi \rrbracket_\gamma^\mathcal{V} = \llbracket \varphi \rrbracket_\gamma^\mathcal{V}$.*

Proof. Since $\mathcal{L}^\mathcal{V} = \bigcup_{n \in \omega} \mathcal{L}_n^\mathcal{V}$, the claim will follow from the commutativity of:

$$\begin{array}{ccc} \mathcal{L}_n^\mathcal{V} & \xrightarrow{\xi_n} P(\mathbb{T}F_X)^n X & \xrightarrow{P(\mathbb{T}F_X)^{n-1} \gamma_X \circ \dots \circ \gamma_X} PX \\ \rho_n \downarrow & \uparrow P((\lambda_X)_n)_X & \nearrow P(\gamma_X)_n \\ PF_X^n X & \xrightarrow{\sigma_{(F_X)^n X}} P\mathbb{T}(F_X)^n X & \end{array}$$

for $n \in \omega$, where the right triangle commutes by Lemma 5.8, and the commutativity of the left rectangle is proved below by induction on n .

The case $n = 0$ follows immediately from $P\eta \circ \sigma = \text{id}$.

Using $\mathcal{L}_{n+1}^\mathcal{V} = \mathcal{V} + L\mathcal{L}_n^\mathcal{V}$, the inductive step follows from the commutativity of the following two diagrams:

$$\begin{array}{ccccccc}
 \mathcal{V} & \xrightarrow{V} & PX & \xrightarrow{P\pi_1} & P(X \times F(\mathbb{T}F_X)^n X) & \xrightarrow{\sigma_{F_X(\mathbb{T}F_X)^n X}} & P(\mathbb{T}F_X)^{n+1} X \\
 \downarrow V & & & & & & \uparrow P((\lambda_X)_{n+1})_X \\
 PX & \xrightarrow{P\pi_1} & PF_X(F_X^n X) & \xrightarrow{\sigma_{F_X^{n+1} X}} & P\mathbb{T}F_X^{n+1} & &
 \end{array}$$

and

$$\begin{array}{ccccccc}
 L\mathcal{L}_n^\mathcal{V} & \xrightarrow{L\xi_n} & LP(\mathbb{T}F_X)^n X & \xrightarrow{(\delta_X)(\mathbb{T}F_X)^n X} & PF_X(\mathbb{T}F_X)^n X & \xrightarrow{\sigma_{F_X(\mathbb{T}F_X)^n X}} & P(\mathbb{T}F_X)^{n+1} X \\
 \downarrow L\rho_n & & \uparrow LP((\lambda_X)_n)_X & & \uparrow PF_X((\lambda_X)_n)_X & & \uparrow P\mathbb{T}F_X((\lambda_X)_n)_X \\
 LPF_X^n X & \xrightarrow{L\sigma_{F_X^n X}} & LP\mathbb{T}F_X^n X & \xrightarrow{(\delta_X)\mathbb{T}F_X^n X} & PF_X\mathbb{T}F_X^n X & \xrightarrow{\sigma_{F_X\mathbb{T}F_X^n X}} & P\mathbb{T}F_X\mathbb{T}F_X^n X \\
 \downarrow (\delta_X)_{F_X^n X} & & & & \uparrow P(\lambda_X)_{F_X^n X} & & \uparrow P\mathbb{T}(\lambda_X)_{F_X^n X} \\
 PF_X^{n+1} X & \xrightarrow{\sigma_{F_X^{n+1} X}} & P\mathbb{T}F_X^{n+1} X & \xrightarrow{\sigma_{\mathbb{T}F_X^{n+1} X}} & P\mathbb{T}^2 F_X^{n+1} X & &
 \end{array}$$

Commutativity of the former diagram is an easy exercise. For the commutativity of the latter, note that the top arrow is $\xi_{n+1} \circ \iota_2$, the top-left rectangle commutes by the induction hypothesis, the top-middle rectangle commutes by naturality of δ_X , the bottom-left rectangle commutes by Corollary 5.14, the top-right and bottom-right rectangles commute by naturality of σ , and finally the long arrow from $L\mathcal{L}_n^\mathcal{V}$ to $P(\mathbb{T}F_X)^{n+1} X$ is $P((\lambda_X)_{n+1})_X \circ \sigma_{F_X^{n+1} X} \circ \rho_{n+1} \circ \iota_2$ as required – the latter follows from:

$$\begin{aligned}
 P\mathbb{T}F_X((\lambda_X)_n)_X \circ P\mathbb{T}(\lambda_X)_{F_X^n X} \circ \sigma_{\mathbb{T}F_X^{n+1} X} \circ \sigma_{F_X^{n+1} X} \circ (\delta_X)_{F_X^n X} \circ L\rho_n &= \text{(Lemma 5.1)} \\
 P\mathbb{T}F_X((\lambda_X)_n)_X \circ P\mathbb{T}(\lambda_X)_{F_X^n X} \circ P\mu_{F_X^{n+1} X} \circ \sigma_{F_X^{n+1} X} \circ (\delta_X)_{F_X^n X} \circ L\rho_n &= \text{(Definition 5.7)} \\
 P((\lambda_X)_{n+1})_X \circ \sigma_{F_X^{n+1} X} \circ (\delta_X)_{F_X^n X} \circ L\rho_n &= \text{(Definition 5.9)} \\
 P((\lambda_X)_{n+1})_X \circ \sigma_{F_X^{n+1} X} \circ \rho_{n+1} \circ \iota_2 &
 \end{aligned}$$

This concludes the proof. \square

The next example confirms that by using τ_\square instead of μ_1 to resolve branching for $\mathbb{T} = \mathbb{T}_S$ with $S = (\{0, 1\}, \vee, 0, \wedge, 1)$, Theorem 5.15 does not hold for functors F with associated linear time modalities of arity 2 or greater.

Example 5.16. Let $S = (\{0, 1\}, \vee, 0, \wedge, 1)$. Assume that τ_\square is used to resolve branching, and consider the following $\mathbb{T}_S \circ (1 + A \times \text{Id} \times \text{Id})$ -coalgebra (X, γ) :

$$\begin{array}{ccccc}
 & & x_1 & \rightsquigarrow b & \longrightarrow x_3 & \rightsquigarrow * \\
 & \nearrow & & \searrow & & \\
 x_0 & \rightsquigarrow a & \longrightarrow x_2 & & x_4 & \rightsquigarrow *
 \end{array}$$

where \rightsquigarrow is used for nondeterministic transitions (and thus x_2 is a deadlock state). Under the step-wise semantics, the formula $[a](*, *)$ does not hold in x_0 , as although x_2 satisfies $*$

(since it has no outgoing transitions), x_1 does not: according to the definition, for a state to satisfy $*$, all transitions from that state (if any) must be terminating ones. However, the map $\gamma_2 : X \rightarrow \mathsf{T}_S(1 + A \times (1 + A \times X) \times (1 + A \times X))$ maps x_0 to the empty set: again, this is because x_2 has no transitions and therefore the flattening performed by γ_2 results in an empty set of linear time behaviours of depth 2; as a result, under the path-based semantics, the formula holds.

We note that the proof of Theorem 5.15 only makes use of the specific (canonical) choice of linear time modalities when it comes to applying Lemma 5.12. As a result, a generalisation of Theorem 5.15 to non-canonical choices for both the linear time modalities and the branching modality can be stated.

Theorem 5.17. *Let $\lambda : F\mathsf{T} \Rightarrow \mathsf{T}F$ be as in Definition 5.4, and let $L : \mathsf{Set} \rightarrow \mathsf{Set}$, $\delta : LP \Rightarrow PF$ and $\sigma : P \Rightarrow PT$ (induced by $\tau : \mathsf{T}^2 1 \rightarrow \mathsf{T}1$) be such that Lemma 5.12 holds. Then the path-based and the step-wise semantics of $\mathcal{L}^\mathcal{V}$ coincide.*

Example 5.18. If $L : \mathsf{Set} \rightarrow \mathsf{Set}$ additionally includes modalities incorporating restricted disjunctions (Example 4.3), the associated $\delta : LP \Rightarrow PF$ together with the canonical choice for $\sigma : P \Rightarrow PT$ satisfy Lemma 5.12, and therefore Theorem 5.17 applies.

5.3. Path-Based Semantics for $\tilde{\mathcal{L}}^{\mathsf{Free}(\mathcal{V})}$. Giving a path-based semantics for $\tilde{\mathcal{L}}^{\mathsf{Free}(\mathcal{V})}$ (which was shown in Section 5.1 to coincide with the fixpoint-free fragment of the logic $\mu\mathcal{L}_A^\mathcal{V}$) can be done in much the same way as for $\mathcal{L}^\mathcal{V}$, since the "logic functor" used to deal with branching is still the identity functor, but now on $\mathsf{Alg}(\mathsf{T})$. For completeness, this section sketches the main definitions and results, all very similar to their counterparts in Section 5.2.

As before, we let $\tilde{\mathcal{L}}_0^{\mathsf{Free}(\mathcal{V})} = \mathsf{Free}(\mathcal{V})$ and $\tilde{\mathcal{L}}_{n+1}^{\mathsf{Free}(\mathcal{V})} = \mathsf{Free}(\mathcal{V}) + \tilde{L}\tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})}$ for $n \in \omega$. This time however, the coproduct is taken in the category $\mathsf{Alg}(\mathsf{T})$. Again, we have $\tilde{\mathcal{L}}^{\mathsf{Free}(\mathcal{V})} = \bigcup_{n \in \omega} \tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})}$.

Definition 5.19 (Path-Based Semantics for $\tilde{\mathcal{L}}^{\mathsf{Free}(\mathcal{V})}$). Let (X, γ) be a $\mathsf{T}F$ -coalgebra, and let $V : \mathcal{V} \rightarrow PX$ be a valuation. Let $\tilde{\rho}_n : \tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})} \rightarrow \tilde{P}F_X^n X$ with $n \in \omega$ be given by:

- $\tilde{\rho}_0 = V^\sharp : \tilde{\mathcal{L}}_0^{\mathsf{Free}(\mathcal{V})} \rightarrow \tilde{P}X$,
- $\tilde{\rho}_{n+1} = [\tilde{P}\pi_1 \circ V^\sharp, (\tilde{\delta}_X)_{F_X^n X} \circ \tilde{L}\tilde{\rho}_n] : \tilde{\mathcal{L}}_{n+1}^{\mathsf{Free}(\mathcal{V})} \rightarrow \tilde{P}F_X^{n+1} X$ for $n \geq 0$:

$$\mathsf{Free}(\mathcal{V}) \xrightarrow{V^\sharp} \tilde{P}X \xrightarrow{\tilde{P}\pi_1} \tilde{P}F_X^{n+1} X \xleftarrow{(\tilde{\delta}_X)_{F_X^n X}} \tilde{L}\tilde{P}F_X^n X \xleftarrow{\tilde{L}\tilde{\rho}_n} \tilde{L}\tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})}$$

where, as before, we write $\tilde{\delta}_X : \tilde{L}\tilde{P} \Rightarrow \tilde{P}F_X$ for the natural transformation given by $\tilde{P}\pi_2 \circ \tilde{\delta}$. Now for $\varphi \in \tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})}$, the *path-based semantics* $\llbracket \varphi \rrbracket_\gamma^{V^\sharp}$ is given by the image of φ under the map

$$\tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})} \xrightarrow{\tilde{\rho}_n} \tilde{P}F_X^n X \xrightarrow{\tilde{\sigma}_{F_X^n X}} \tilde{P}\mathsf{T}F_X^n X \xrightarrow{\tilde{P}(\gamma_X)_n} \tilde{P}X$$

Definition 5.20. Let $V : \mathcal{V} \rightarrow PX$ be a valuation. For $n \geq 1$, let $\tilde{\xi}_n : \tilde{\mathcal{L}}_n^{\mathsf{Free}(\mathcal{V})} \rightarrow \tilde{P}(\mathsf{T}F_X)^n X$ be defined by:

- $\tilde{\xi}_0 = V^\sharp : \mathsf{Free}(\mathcal{V}) \rightarrow \tilde{P}X$,

- $\tilde{\xi}_{n+1} = \tilde{\sigma}_{F_X(\mathbb{T}F_X)^n X} \circ [\tilde{P}\pi_1 \circ V^\sharp, (\tilde{\delta}_X)_{(\mathbb{T}F_X)^n X} \circ \tilde{L}\tilde{\xi}_n]$ for $n \geq 1$.

Lemma 5.21. *For formulas in $\tilde{\mathcal{L}}_n^{\text{Free}(\mathcal{V})}$, the step-wise semantics is obtained by post-composing $\tilde{\xi}_n$ with $\tilde{P}\gamma_X \circ \dots \circ \tilde{P}(\mathbb{T}F_X)^{n-1}\gamma : \tilde{P}(\mathbb{T}F_X)^n X \rightarrow PX$.*

The next lemma allows Theorem 5.15 to be lifted to the logics $\tilde{\mathcal{L}}^{\text{Free}(\mathcal{V})}$.

Lemma 5.22. *Let $\delta : LP \Rightarrow PF$, $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ and $\sigma : P \Rightarrow P\mathbb{T}$ be as in Lemma 5.12, and let $\tilde{\delta} : \tilde{L}\tilde{P} \Rightarrow \tilde{P}F$ and $\tilde{\sigma} : \tilde{P} \Rightarrow \tilde{P}\mathbb{T}$ arise from δ and σ as before. Then the following diagram commutes:*

$$\begin{array}{ccc}
 \tilde{L}\tilde{P} & \xrightarrow{\tilde{L}\tilde{\sigma}} & \tilde{L}\tilde{P}\mathbb{T} \xrightarrow{\tilde{\delta}_{\mathbb{T}}} \tilde{P}F\mathbb{T} \\
 \tilde{\delta} \Downarrow & & \Uparrow \tilde{P}\lambda \\
 \tilde{P}F & \xrightarrow{\tilde{\sigma}_F} & \tilde{P}\mathbb{T}F
 \end{array} \tag{5.5}$$

Proof. By freeness of $\tilde{L}\tilde{P}$, it suffices to show that pre-composing the image under U of the above diagram with $\eta_{LU\tilde{P}}$ commutes in **Set**:

$$\begin{array}{ccccc}
 LP & \xlongequal{\quad} & LU\tilde{P} & \xrightarrow{LU\tilde{\sigma}=L\sigma} & LU\tilde{P}\mathbb{T} \\
 \delta \Downarrow & & \eta_{LU\tilde{P}} \Downarrow & & \eta_{LU\tilde{P}\mathbb{T}} \Downarrow \\
 & & U\text{Free}LU\tilde{P} & \xrightarrow{U\text{Free}LU\tilde{\sigma}} & U\text{Free}LU\tilde{P}\mathbb{T} \xrightarrow{U\tilde{\delta}_{\mathbb{T}}} U\tilde{P}F\mathbb{T} \\
 & & U\tilde{\delta} \Downarrow & & \Uparrow U\tilde{P}\lambda=P\lambda \\
 PF & \xlongequal{\quad} & U\tilde{P}F & \xrightarrow{U\tilde{\sigma}_F=\sigma_F} & U\tilde{P}\mathbb{T}F
 \end{array}$$

This, in turn, is a direct consequence of Lemma 5.12. \square

Theorem 5.23. *Let $\delta : LP \Rightarrow PF$, $\lambda : F\mathbb{T} \Rightarrow \mathbb{T}F$ and $\sigma : P \Rightarrow P\mathbb{T}$ be as in Lemma 5.22. Also, let (X, γ) be a $\mathbb{T}F$ -coalgebra and let $V : \mathcal{V} \rightarrow PX$ be a valuation. For $\varphi \in \tilde{\mathcal{L}}^{\text{Free}(\mathcal{V})}$, $\llbracket \varphi \rrbracket_\gamma^{V^\sharp} = \llbracket \varphi \rrbracket_\gamma^{V^\sharp}$.*

Proof. Exactly the same as the proof of Theorem 5.15, except that Lemma 5.22 is used instead of Lemma 5.12. \square

6. A GENERALISATION

The logics described in Section 4 apply to coalgebras of type $\mathbb{T}_S \circ F$. However, they can easily be generalised to coalgebraic types $F_1 \circ \mathbb{T}_S \circ \dots \circ F_{n-1} \circ \mathbb{T}_S \circ F_n$ obtained as finite compositions of *arbitrary* endofunctors F_1, \dots, F_n and a single partial semiring monad \mathbb{T}_S . The domain of truth values is still S , and the semantics of formulas makes use of the extension predicate lifting ext and of predicate liftings for each of F_1, \dots, F_n , as required by the coalgebraic type. Rather than making explicit such a generalisation, this section shows that this has useful instantiations. Our first instantiation yields a logic for resource-aware, potentially infinite computations, where resources can be replenished at certain points in the computation. Our second instantiation results in a logic for component interaction, in a setting where the behaviour of different components is modelled using different branching

monads; this is problematic since the composition of two (weighted) monads is not usually a monad (yet alone a weighted one).

To cover the above two examples, we consider system types where the "ingredient" functors (the F_i s and the monad \mathbb{T}_S above) play three different rôles:

- Some of these endofunctors (which we denote by F_1, F_2, \dots) describe the type of linear behaviour the system exhibits in any particular run.
- A semiring monad \mathbb{T}_S describes the (main) branching type. This is used to derive the domain of truth values (namely S), and provides a canonical way to abstract away \mathbb{T}_S -branching (through the canonical extension lifting ext).
- Other endofunctors (which we denote by G_1, G_2, \dots) describe information that is to be abstracted away in a not necessarily canonical way, using a *single* predicate lifting for each occurrence of such an endofunctor in the coalgebraic type. We note that a similar approach is taken in [KR15], where branching is abstracted away using arbitrary (as opposed to canonical) predicate liftings.

In particular, the last point means that the logics can have a similar syntax as before, with modal operators arising from the structure of the F_i s only. However, at the semantic level, additional predicate liftings are involved, corresponding to the presence of the G_i s in the coalgebraic type.

Our approach applies to arbitrary compositions of endofunctors of the three types identified above. However, for simplicity of presentation, we restrict to $G \circ \mathbb{T}_S \circ F$ -coalgebras in what follows. With this restriction, our syntax remains single-sorted, whereas in the general case a multi-sorted syntax similar to that of [CP07] would be required. As running examples we consider:

- (1) $(S \times \text{Id}) \circ \mathbb{T}_S \circ F$ -coalgebras, with $S : \text{Set} \rightarrow \text{Set}$ the constant functor $X \mapsto S$. Thus, $G = S \times \text{Id}$. Such coalgebras model weighted systems (with weights being thought of as costs) with linear behaviour specified by F , some of whose states (those for which the first component of the coalgebra map returns a non-zero value) allow for resource replenishment (according to the specified value).
- (2) $\mathbb{T}_{S'} \circ \mathbb{T}_S \circ F$ -coalgebras, with $S' := (S', +', 0', \bullet', 1')$ another partial commutative semiring. Thus, $G = \mathbb{T}_{S'}$. Such coalgebras can model interactions between components modelled as $\mathbb{T}_{S'} \circ F_1$ - and respectively $\mathbb{T}_S \circ F_2$ -coalgebras, with the functor F being obtained as a kind of "synchronous product" of F_1 and F_2 . For example, if $F_1 = \text{Id} \times A$ models systems with outputs (in A) and $F_2 = (B \times \text{Id})^A$ models systems with inputs (from A) and outputs (in B), a natural choice for F is $B \times \text{Id}$. In this case, a $\mathbb{T}_{S'} \circ F_1$ -coalgebra (C, γ) and a $\mathbb{T}_S \circ F_2$ -coalgebra (D, δ) combine to yield a $\mathbb{T}_{S'} \circ \mathbb{T}_S \circ F$ -coalgebra, using the strength maps of \mathbb{T}_S and $\mathbb{T}_{S'}$, and a mapping $\text{sync} : F_1 X \times F_2 Y \rightarrow F(X \times Y)$, natural in X and Y , which models synchronisation between F_1 - and F_2 -behaviours:

$$\begin{array}{c}
 C \times D \xrightarrow{\gamma \times \delta} \mathbb{T}_{S'} F_1 C \times \mathbb{T}_S F_2 D \xrightarrow{\text{st}_{\mathbb{T}_{S'}}} \mathbb{T}_{S'} (F_1 C \times \mathbb{T}_S F_2 D) \xrightarrow{\mathbb{T}_{S'} \text{st}_{\mathbb{T}_S}} \mathbb{T}_{S'} \mathbb{T}_S (F_1 C \times F_2 D) \\
 \downarrow \mathbb{T}_{S'} \mathbb{T}_S \text{sync} \\
 \mathbb{T}_{S'} \mathbb{T}_S F(C \times D)
 \end{array}$$

where $\text{sync} : F_1 C \times F_2 D = (C \times A) \times (B \times D)^A \rightarrow B \times (C \times D) = F(C \times D)$ is given by $\text{sync}(c, a, f) = (b, (c, d))$ if $f(a) = (b, d)$, for $c \in C$, $a \in A$ and $f \in (B \times D)^A$.

For $F = \coprod_{\lambda \in \Lambda} X^{\text{ar}(\lambda)}$, our logic will again use the canonical predicate liftings $\llbracket \lambda \rrbracket$. In addition, we fix a *monotone* predicate lifting $\text{ext}^G : S^- \Rightarrow S^{G,-}$, to be used to abstract away G -behaviour. (Note that when $G = \mathsf{T}_{S'}$, the predicate lifting ext^G acts on S -valued predicates.) The syntax of the logics is as before, whereas the semantics now interprets modal operators by:

$$\llbracket [\lambda](\varphi_1, \dots, \varphi_{\text{ar}(\lambda)}) \rrbracket_\gamma^V = \gamma^*(\text{ext}_{\mathsf{T}_S F X}^G(\text{ext}_{F X}(\llbracket \lambda \rrbracket_X(\llbracket \varphi_1 \rrbracket_\gamma^V, \dots, \llbracket \varphi_{\text{ar}(\lambda)} \rrbracket_\gamma^V))))$$

for a $G \circ \mathsf{T}_S \circ F$ -coalgebra $\gamma : X \rightarrow G\mathsf{T}_S F X$. In particular, this definition also ensures the monotonicity of the operators used to interpret fixpoint formulas. We also note that, while the interpretation of propositional operators (formal weighted sums) remains unchanged, the resulting semantics is different, as the actual weighted sums are now computed *after* the application of ext^G . One might expect the result to be the same as when ext^G is applied after the computation of the weighted sums, however, this is only true when all the weights are equal to 1, or when the predicate liftings ext^G and ext commute. The latter condition fails to hold in our second example below, and therefore formal weighted sums with the weights different from 1 should be used with care in that case.

It is also worth noting that the results of Section 5 do not apply to the second of these instantiations, as in this case the predicate liftings ext^G and ext do not commute. Nevertheless, the second logic also has a linear time flavour (with F describing linear time behaviour), which in this case can *only* be captured using a step-wise semantics.

6.1. A Logic for Resources. The logics obtained by taking $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$ allow reasoning about the minimal cost of exhibiting a given linear behaviour in states of $\mathsf{T}_S \circ F$ -coalgebras. This has limited usefulness, as infinitely running systems will typically incur an infinite cost to achieve a given (infinite) behaviour. This section shows how suitably instantiating the approach outlined above can address this issue.

We begin by noting that incorporating resource replenishment into the computational monad T_S would require negative costs, which would in turn lead to technical difficulties caused by lack of monotonicity of the operators used to interpret fixpoint formulas. Similarly, modelling resource replenishment as part of the linear behaviour would alter the associated notion of trace, if indeed this was possible using only canonical predicate liftings.

Here we take a different approach, namely moving to coalgebras of type $S \times \mathsf{T}_S \circ F$ -coalgebras, where the first component in the coalgebra structure provides, for each state, the amount of resources made available when visiting that state. Thus, $G = S \times \text{Id}$, and as associated predicate lifting $\text{ext}^{S \times \text{Id}} : S^{\text{Id}} \Rightarrow S^{S \times \text{Id}}$ we take:

$$\text{ext}^{S \times \text{Id}}(p)(s, x) = p(x) \ominus s$$

for $p : X \rightarrow S$, where $\ominus : S \times S \rightarrow S$ is given by:

$$n \ominus m = \text{supp}\{z \mid m + z \sqsubseteq n\}$$

The operation \ominus is thus a kind of subtraction operation, capped below at 0 (the smallest element in S). In particular, if there exists no $z \in S$ such that $m + z \sqsubseteq n$, then $n \ominus m = 0$. For example, if $S = (\mathbb{N}^\infty, \min, \infty, +, 0)$, we have:

$$n \ominus m = \begin{cases} \max(n - m, 0), & \text{if } m \neq \infty \text{ or } n \neq \infty, \\ \infty, & \text{otherwise.} \end{cases}$$

and the resulting logic can be used to reason about minimal costs in a setting where new resources can be made available as the computation proceeds.

The above choice of $\text{ext}^{S \times \text{Id}}$ yields the following interpretation of modal formulas:

$$\llbracket [\lambda](\varphi_1, \dots, \varphi_{\text{ar}(\lambda)}) \rrbracket_\gamma^V = \gamma_2^*(\text{ext}_{FX}(\llbracket [\lambda]_X(\llbracket \varphi_1 \rrbracket_\gamma^V, \dots, \llbracket \varphi_{\text{ar}(\lambda)} \rrbracket_\gamma^V) \rrbracket)) \ominus \gamma_1$$

for each $S \times \mathsf{T}_S \circ F$ -coalgebra $\gamma = \langle \gamma_1, \gamma_2 \rangle : X \rightarrow S \times \mathsf{T}_S F X$. As a result, states x where resources are being replenished ($\gamma_1(x) \neq 0$), have their *future* costs offset by $\gamma_1(x)$.

Since both $\text{ext}^{S \times \text{Id}}$ and the predicate liftings $\llbracket [\lambda] \rrbracket$ commute with ext , a path-based semantics for the resulting logic can be defined similarly to Section 5, and proved equivalent to the step-wise semantics. In the path-based semantics, finite-depth weighted branching arising from successive unfoldings of the $G \circ \mathsf{T}_S \circ F$ -coalgebra structure is flattened into a weighted sum of "paths" of type $G \circ F$. Therefore in this case the qualification "linear time" is fully justified for the resulting logic.

6.2. A Logic for Component Interaction. Taking $G = \mathsf{T}_{S'}$ with $S' = (\{0, 1\}, \vee, 0, \wedge, 1)$, with associated predicate lifting $\text{ext}^G : S^{\text{Id}} \Rightarrow S^{(S'^{\text{Id}})}$ given by:

$$\text{ext}_X^G(p)(p') = \max\{p(x) \mid p'(x) = 1\}, \text{ for } p : X \rightarrow S \text{ and } p' : X \rightarrow \{0, 1\}.$$

yields a logic for reasoning about the interaction between a weighted system and a non-deterministic adversary. If weights model costs, then the semantics of the logic measures, for each formula $\varphi \in \mu\mathcal{L}_\Lambda$, the minimal cost of conforming to φ , *irrespective of the behaviour of the adversary*. As previously mentioned, in this case ext^G and ext do not commute, and therefore the resulting logic is not, strictly speaking, a linear time logic.

7. CONCLUSIONS AND FUTURE WORK

We have described a uniform approach to defining linear time fixpoint logics for systems that incorporate branching behaviour, and have proved the equivalence of the step-wise semantics of these logics with an alternative path-based semantics, akin to those employed by standard linear time logics. While this equivalence result motivates the use of the term "linear time" to describe (the canonical versions of) our logics, the step-wise semantics is more appealing in its simplicity, uniformity and direct connection to automata on infinite structures.

We have also showed that allowing for some arbitrary (rather than all canonical) choices for the predicate liftings employed in the semantics of the logics increases their applicability, while maintaining a certain linear time flavour. We believe that part of the value of the present paper is to allow one to reason systematically about the precise linear time nature of such logics.

Ongoing work concerns the study of the connection between (the canonical versions of) our linear time logics and automata, as well as of the expressiveness of these logics. Preliminary results on an automata-theoretic approach to model checking have already been obtained [CSH], but further work is needed to relax the current assumptions on the underlying semiring (which require strictly ascending/descending chains in the partial order (S, \sqsubseteq) to be bounded in length).

REFERENCES

- [BK05] M.M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *Foundations of Software Science and Computational Structures - 8th International Conference, FOSSACS 2005, Proceedings*. Springer, 2005.
- [BK08] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.
- [Cîr14] C. Cîrstea. A coalgebraic approach to linear-time logics. In *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Proceedings*, pages 426–440, 2014.
- [Cîr15] C. Cîrstea. Canonical coalgebraic linear time logics. In *6th Conference on Algebra and Coalgebra in Computer Science, CALCO 2015, Proceedings*, pages 66–85, 2015.
- [Cîr17] C. Cîrstea. From branching to linear time, coalgebraically. *Fundamenta Informaticae*, 150:1–28, 2017.
- [CJ13] D. Coumans and B. Jacobs. Scalars, monads, and categories. In C. Heunen, M. Sadrzadeh, and E. Grefenstette, editors, *Quantum Physics and Linguistics. A Compositional, Diagrammatic Discourse*, pages 184–216. Oxford Univ. Press, 2013.
- [CKP11] C. Cîrstea, C. Kupke, and D. Pattinson. Exptime tableaux for the coalgebraic μ -calculus. *Logical Methods in Computer Science*, 7(3), 2011.
- [CP07] C. Cîrstea and D. Pattinson. Modular construction of complete coalgebraic logics. *Theor. Comput. Sci.*, 388(1-3):83–108, 2007.
- [CSH] C. Cîrstea, S. Shimizu, and I. Hasuo. Parity automata for quantitative linear time logics. Draft.
- [Dam92] M. Dam. Fixed points of Büchi automata. In R. Shyamasundar, editor, *Foundations of Software Technology and Theoretical Computer Science, Proceedings*, pages 39–50. Springer, 1992.
- [DP02] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order (2. ed.)*. Cambridge University Press, 2002.
- [DR06] M. Droste and G. Rahonis. Weighted automata and weighted logics on infinite words. In O.H. Ibarra and Z. Dang, editors, *Developments in Language Theory: 10th International Conference, DLT 2006, Proceedings*. Springer, 2006.
- [GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
- [Has15] I. Hasuo. Generic weakest precondition semantics from monads enriched with order. *Theor. Comput. Sci.*, 604:2–29, 2015.
- [HJ98] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.
- [Jac16] B. Jacobs. *Introduction to Coalgebra. Towards Mathematics of States and Observations*. Cambridge University Press, 2016.
- [Koc12] A. Kock. Commutative monads as a theory of distributions. *Theory and Applications of Categories*, 26(4):97–131, 2012.
- [KR15] B. Klin and J. Rot. Coalgebraic trace semantics via forgetful logics. In *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Proceedings*, pages 151–166, 2015.
- [Mei09] I. Meinecke. A weighted μ -calculus on words. In *Proceedings of the 13th International Conference on Developments in Language Theory, DLT '09*, pages 384–395. Springer, 2009.
- [Pat03] D. Pattinson. Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theor. Comput. Sci.*, 309(1-3):177–193, 2003.
- [Ven12] Y. Venema. Lectures on the modal μ -calculus. Lecture notes, Institute for Logic, Language and Computation, University of Amsterdam, 2012.